Ruslan Padnevych

Licenciado em Engenharia Informática

SmartyFlow - Biometria Facial Robusta para Identificação Virtual

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática

Orientador: Doutor João Miguel da Costa Magalhães, Professor Associado,
Universidade Nova de Lisboa

Co-orientador: David Fernandes Semedo, Investigador Auxiliar Convidado,
Universidade Nova de Lisboa

Júri

Presidente: Doutor Carlos Augusto Isaac Piló Viegas Damásio
Arguente: Doutor Rui Manuel Feliciano de Jesus

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2021

SmartyFlow - Biometria Facial Robusta para Identificação Virtual

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2]. [1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Agradecimentos

Gostaria de deixar aqui os meus sinceros agradecimentos a todos os envolvidos para que o desenvolvimento desta dissertação fosse possível.

Em primeiro lugar, quero agradecer à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, que me acolheu durante 5 anos, e que me fez crescer a nível profissional e pessoal.

Em segundo lugar, agradeço ao meu orientador João Magalhães, por me guiar ao longo do ano, pelo apoio, pelas observações críticas, e pelas discussões que ajudaram a definir o caminho para o desenvolvimento deste trabalho. Quero também agradecer a colaboração do meu co-orientador, David Semedo, que me acompanhou e auxiliou bastante nas implementações ao transmitir o seu conhecimento. Por fim, agradeço a *VisionBox* por ter financiado uma bolsa ao longo do desenvolvimento desta tese, no âmbito do projeto *SmartyFlow*, Programa Operacional Regional de Lisboa, PT2020 (LISBOA-01-0247-FEDER-017283).

Aproveito ainda para agradecer aos meus colegas e amigos que levo para a vida Alexandre Jacinto, Álvaro Soares, Bruno Carvalho, Diogo Nunes, Diogo Romão, Edgar Silva e Pedro Rodrigues pelo apoio desde o primeiro dia do meu percurso académico e por todas as observações que me permitiram aperfeiçoar esta dissertação.

Os meus agradecimentos, mas não menos importantes, são aos meus pais, a minha irmã e aos familiares da Ucrânia, que sempre acreditaram no sucesso da minha carreira académica e no meu desenvolvimento pessoal. E por último a Mafalda, pela amizade e amor, e pela paciência de ouvir as minhas preocupações e alegrias que esta dissertação me proporcionou.

“ If you don’t design your own life plan, chances are you’ll fall into someone else’s plan. And guess what they have planned for you? Not much. ” (Jim Rohn)

Resumo

O roubo de identidade é um problema crescente na nossa sociedade em geral. Deste modo, é necessário garantir que os métodos de autenticação existentes sejam seguros contra ataques de apresentação. Nesta tese pretende-se estudar métodos de autenticação com base em biometria facial, mais especificamente, verificação facial. Trata-se de um método que, apesar de moderno, é igualmente vulnerável a ataques de segurança, em particular ataques de falsificaçãodo rosto. Ultimamente, têm surgido abordagens que utilizam a verificação da vivacidade para detetar tais ameaças.

Assim, no contexto desta tese, a vivacidade será detetada através de um vídeo da face de um indivíduo, utilizando o seu ritmo cardíaco estimado através de *Eulerian Video Magnification (EVM)*. Ritmo cardíaco este que é posteriormente classificado recorrendo a dois tipos de redes neurais profundas diferentes: *Convolution Neural Network (CNN)* e *Temporal Convolutional Network (TCN)*. Utilizando esta técnica de deteção, é possível garantir maior resiliência a ataques de apresentação, pois o ritmo cardíaco é uma característica fisiológica dificilmente falsificável.

Para além de classificar o sinal do ritmo cardíaco estimado, procurou-se desenvolver uma forma eficiente de melhorar ainda mais a robustez dos modelos implementados ao detetar os ataques de apresentação. Para isso, com base no Treino Adversarial desenvolveu-se a *Deep Convolutional Generative Adversarial Network (DCGAN)* que permite a criação de sinais cardíacos artificiais.

Como resultado concluiu-se que a rede *TCN* é mais apropriada para esta tarefa (obtendo 90,17 de eficácia sem sinais artificiais) e que a introdução de sinais artificiais produzidos pela *DCGAN* permitem de facto melhorar a robustez do modelo (obtendo 93,55 de eficácia).

Palavras-chave: Batimento cardíaco; Biometria facial; *CNN*; Deteção de vivacidade; Falsificação do rosto; *GAN*; *TCN*; Técnicas anti-falsificação facial

Abstract

Identity theft is an ever-increasing problem in our society. Thus, it is necessary to ensure that the existing authentication methods are secure against presentation attacks. The proposed thesis aims to study authentication methods based on facial biometrics, more specifically, facial verification. Nonetheless, despite being a rather modern method, it is also vulnerable to security attacks, in particular, to face spoofing. Several approaches have recently emerged that use liveness checks to detect such threats.

So, in the context of this thesis, liveness will be detected through a video of an individual's face, using its estimated heart rate estimated through *EVM*. The heart rate is then classified using two different types of deep neural networks: *CNN* e *TCN*. By using this detection technique, it is possible to ensure a higher level of resilience to presentation attacks, considering that heart rate is a physiological characteristic that is difficult to forge.

Besides classifying the estimated heart rate signal, an efficient way to increase the robustness of the implemented models in detecting presentation attacks was developed. To achieve this, on the basis of Adversarial Training, the *DCGAN* was developed, which allows the creation of artificial heart signals.

As a result it was concluded that the *TCN* is more appropriate for this task (achieving 90,17 efficacy without artificial signals) and that the introduction of artificial signals produced by *DCGAN* can in fact improve the robustness of the model (achieving 93,55 efficacy).

Keywords: CNN; Face anti-spoofing techniques; Face spoofing; Facial biometrics; GAN; Heart rate; Liveness detection; TCN

Índice

Lista de Figuras

Lista de Tabelas

Siglas

AUC Area Under the Curve

BCELoss Binary Cross Entropy Loss

CNN Convolution Neural Network

DCGAN Deep Convolutional Generative Adversarial Network

DMD Dynamic Mode Decomposition

EUSIPCO European Signal Processing Conference

EVM Eulerian Video Magnification

FN False Negative

FNR False Negative Rate

FP False Positive

FPR False Positive Rate

GAN Generative Adversarial Network

IEC International Electrotechnical Commission

ISO International Organization for Standardization

LTSS Long-term Statistical Spectral

ML Machine Learning

PAD Presentation Attack Detection

PAI Presentation Attack Instrument

ROC Receiver Operator Characteristic

rPPG Remote Photoplethysmography signal

SVM Support Vector Machine

TCN Temporal Convolutional Network

TN True Negative

TNR True Negative Rate

TP True Positive

TPR True Positive Rate

Capítulo 1

Introdução

1.1 Contexto

Hoje em dia, praticamente todas as pessoas utilizam diariamente algum tipo de tecnologia ou dispositivo que fornece um conjunto de funcionalidades inteligentes. A sua utilização facilita bastante imensas tarefas do quotidiano, o que lhes permite poupar muito tempo. No entanto, algumas destas tecnologias requerem acesso à informação que é considerada pessoal, para o seu devido funcionamento. Dito isto, cabe a cada indivíduo decidir que tipo de informação está disposto a partilhar. Atualmente, é necessário ter um cuidado especial com os dados que se partilham na Internet, uma vez que, podem cair na posse das pessoas erradas e serem utilizados para efetuar ataques a diversos sistemas através do roubo de identidade.

Perante estes riscos, são várias as empresas que têm preferido utilizar nos seus sistemas o método de autenticação mais recente que é o reconhecimento facial. Apesar de mais seguro, as falhas existentes neste processo de autenticação são uma realidade, não excluindo por completo o risco de ocorrer uma identificação errada. Ataques como falsificação do rosto são habitualmente aplicados para ultrapassar um sistema de segurança baseado em reconhecimento ou verificação facial. Um desses exemplos aconteceu em 2011, quando um passageiro embarcou no avião em Hong Kong utilizando uma máscara de um idoso e conseguindo, com sucesso, aterrar no Canadá Figura 1.1[1]. Assim,

[1] https://www.dailymail.co.uk/news/article-1326885/Man-boards-plane-disguised-old-ma

torna-se crucial detetar tais ataques realizados por criminosos, incluindo terroristas que podem causar graves danos na sociedade.

Figura 1.1: Exemplo de um **ataque de apresentação** bem sucedido.

Dado o exemplo apresentado acima, segundo a *International Organization for Standardization* [39] é considerado como sendo um ataque de apresentação quando o impostor interage com o subsistema de recolha de dados biométricos com o objetivo de interferir na operação do sistema biométrico.

O exemplo real mencionado acima, justifica o crescente protagonismo que o reconhecimento e a verificação facial está a obter no controlo de fronteiras em aeroportos. Para além da segurança acrescida, permitem agilizar a passagem em controlos de fronteira. Por exemplo, permitir que os passageiros "autorizados" por tecnologia de verificação facial, possam atravessar o controlo de fronteiras sem paragens como foi demonstrado pelo primeiro protótipo do projeto *SmartyFlow* no Aeroporto de Lisboa em dezembro de 2019, Figura 1.2.

Figura 1.2: Fases do sistema de SmartyFlow: (a) fase inicial do sistema; (b) reconhecimento e verificação facial; (c) desbloqueio da passagem.

n-arrested-arrival-Canada.html

1.2 Âmbito

A tecnologia de *Biometrics on the Move*, permite agilizar o movimento de passageiros em aeroportos eliminando a necessidade de filas de espera para mostrar os documentos aos guardas de fronteira. As vantagens são inúmeras, e não só para os passageiros: os guardas têm mais tempo para efetuar verificações de segurança de forma sistemática e eficiente sem afetar os viajantes regulares, aumentando assim, a segurança nas fronteiras [51].

Figura 1.3: Exemplo do sistema *SmartyFlow* na Fase de Registo: (a) deteção de vivacidade(vídeo de 5s); (b) registo do passaporte; (c) leitura do chipe do passaporte; (d) verificação da correspondência.

A Figura 1.3 e o diagrama da Figura 1.4 permitem visualizar as duas fases de utilização do sistema: a **Fase de Registo** e a **Fase de Autenticação**. O procedimento de identificação e autenticação tem um conjunto de etapas que é necessário satisfazer. Na primeira fase, que é a **Fase de Registo**, o indivíduo precisa previamente de se registar no sistema passando pelos seguintes passos:

- **Captura do vídeo de registo:** a criação de um perfil é feita utilizando uma **Aplicação móvel** como é possível visualizar na Figura 1.4 no ponto (1). Durante a criação exige-se que o indivíduo grave, utilizando a câmara do próprio dispositivo, um vídeo de cinco segundos da sua face neutra numa posição frontal.

- **Deteção de ataque de apresentação:** seguidamente, o vídeo é enviado para o **Sistema de Decisão**, demonstrado em (2) na Figura 1.4, que corresponde à contribuição desta tese. Este sistema tem como objectivo decidir se a face presente no vídeo é de um indivíduo real ou se se trata de um ataque de falsificação facial.

- **Registo da face no sistema:** caso o indivíduo do vídeo seja considerado como sendo autêntico, a melhor face será selecionada e guardada num **Banco de dados**,

como está representado em (3) na Figura 1.4. Desta forma, termina-se a fase de registo. Caso contrário, a aplicação irá solicitar outro vídeo e repetir todo o processo de decisão novamente.

A Figura 1.4 permite enquadrar a contribuição desta tese: iremos focar no sistema de **Deteção de Ataques de Apresentação** e **Registo da face**, descritos pelos dois últimos passos da lista acima.

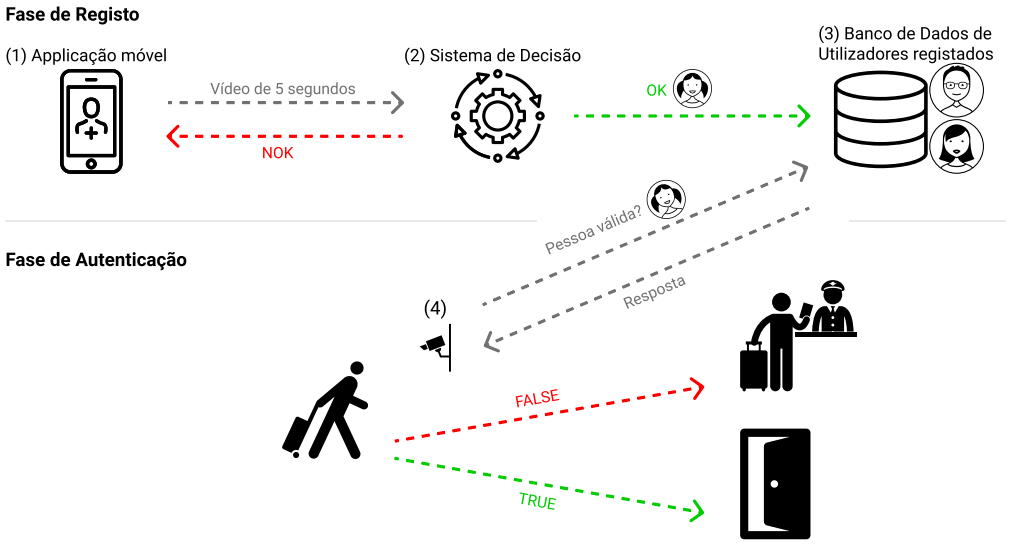


Figura 1.4: Diagrama de caso de uso – processo de registo e autenticação ao sistema de controlo de fronteiras nos aeroportos.

Uma vez registado, o indivíduo pode usufruir do sistema de controlo de fronteiras *SmartyFlow* que elimina barreiras físicas. Assim, na **Fase de Autenticação**, será feita a utilização dos dados biométricos registados no sistema:

- **Deteção de passageiro:** à medida que o indivíduo vai progredindo pelo corredor de controlo de fronteira, as câmaras que estão apresentadas como sendo a referência (4) na Figura 1.4, vão captando a sua face. Uma vez localizada, é feita uma pesquisa na Base de dados para verificar se a pessoa com a dada face se encontra registada no sistema.

- **Verificação válida:** caso o resultado seja positivo, assinalado na figura como *TRUE*, a pessoa pode avançar no controlo de fronteira sem ter que parar para mostrar os seus documentos, pois a identificação já foi realizada com êxito.

- **Verificação inválida:** caso o resultado seja negativo, *FALSE*, o passageiro será encaminhado para o tradicional controlo de fronteira e balcão de segurança, de modo a realizar a sua identificação e verificação dos documentos.

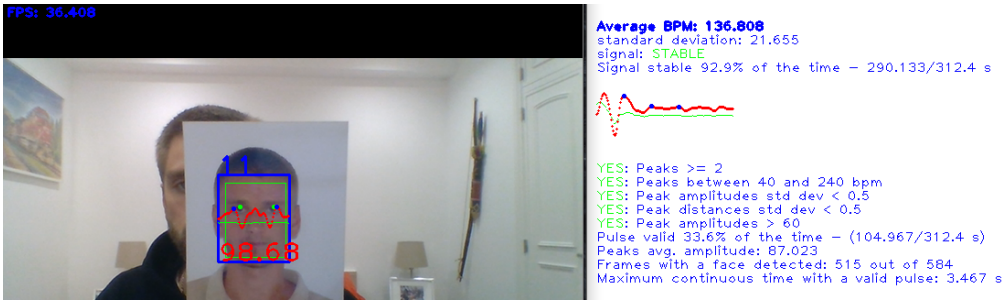
Para concretizar esta funcionalidade para o utilizador, é necessário garantir que o sistema a desenvolver na presente tese seja resiliente a possíveis ataques de apresentação. Apenas assim é possível assegurar que todos os algoritmos que compõem o sistema sejam robustos o suficiente para prevenir roubos de identidade, contribuindo assim, para a preservação da segurança de toda a sociedade.

Desta forma, no âmbito desta tese, pretende-se desenvolver métodos de deteção de ataques a sistemas de verificação facial. A decisão sobre se um dado indivíduo, que esteja a passar o controlo de fronteiras no aeroporto está autorizado ou não, pode ser atacada de várias formas. **No âmbito desta tese endereçamos ataques a sistemas de verificação facial, que ocorram na fase de registo da face do passageiro**. Para tornar o sistema robusto a ataques, serão considerados algoritmos de análise de vídeo que determinem se estão perante um ataque de apresentação ou não. Estes algoritmos de deteção da vivacidade são uma forma de detetar ataques de apresentação. A vivacidade pode ser inferida através do ritmo cardíaco, micro expressões, ou outros indicadores que reflitam o "estado de estar vivo evidenciado através das características anatómicas, reações involuntárias ou funções fisiológicas, reações voluntárias ou comportamentos do sujeito" [39].

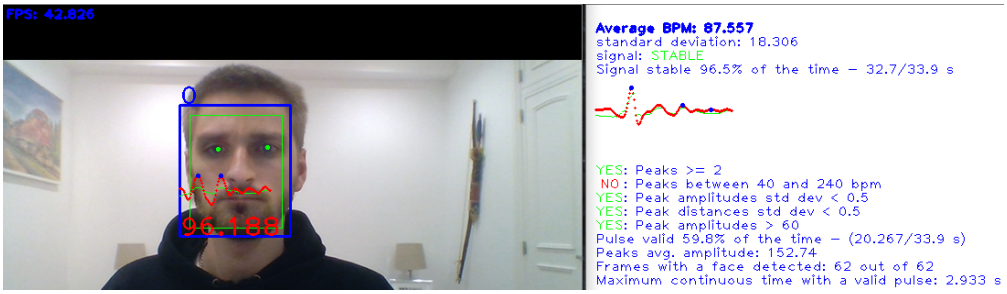
1.3 Objetivos

O objetivo desta tese foca-se sobretudo no desenvolvimento de um **Sistema de Decisão**, que se encontra representado na Figura 1.4 em (2), cujo principal objectivo é ser robusto a ataques do tipo falsificação do rosto. Sistema este que irá utilizar redes neurais profundas, que se têm mostrado bastante eficazes em tarefas de 1D, para determinar a vivacidade do indivíduo e com base nessa informação tomar uma decisão, se o sistema está perante uma falsificação ou não. Para aumentar ainda mais a robustez do classificador aplicou-se uma abordagem chamada *DCGAN* para que fossem gerados variados ataques falsos e utilizados no treino das redes neurais. Uma vez que, é importante garantir que a pessoa que está a registar-se está realmente presente no momento e no local da solicitação [44]. Em relação à estrutura do sistema de **Deteção de Ataques de Apresentação**, estabeleceu-se que será repartido em dois componentes, tal como ilustrado na Figura 1.6:

- **Detecção de ataque de apresentação:** na primeira fase, o processo recebe como *input* um vídeo de cinco segundos enviado a partir do dispositivo do utilizador. Processando o vídeo, extraem-se várias propriedades utilizando os algoritmos já desenvolvidos no âmbito do projecto *SmartyFlow*. Nomeadamente, é estimado o batimento cardíaco, com a ajuda da *EVM framework* para determinar a vivacidade do indivíduo, conforme está simbolizado na Figura 1.6 no ponto (1) e na Figura 1.5. Se o algoritmo implementado na presente tese considerar que o batimento cardíaco estimado a partir da face do indivíduo que está no vídeo é falso (um ataque ao sistema Figura 1.5a), será, então, solicitado ao utilizador que grave outro vídeo. Caso a face seja considerada como real (Figura 1.5b) isto é, batimento cardíaco com sinais de vivacidade, o sistema passará para a segunda fase.



(a)



(b)

Figura 1.5: Utilização do *software* desenvolvido no âmbito do projeto *SmartyFlow*: (a) ataque de apresentação utilizando uma fotografia; (b) indivíduo real.

- **Registo de face:** nesta última etapa, representada na Figura 1.6 em (2), outro algoritmo terá de determinar qual é a face mais neutra e frontal que será, posteriormente, guardada na Base de dados. Esta face irá servir para efetuar futuras autenticações, de forma automática, num sistema de controlo de fronteiras nos aeroportos.

Para cada vídeo recebido como *input* do sistema, caso o indivíduo do vídeo seja real, é necessário selecionar a melhor *frame*, ou seja, a *frame* em que a face do indivíduo é neutra e está com uma orientação frontal. Esta seleção será feita utilizando algoritmos de deteção de expressões faciais.

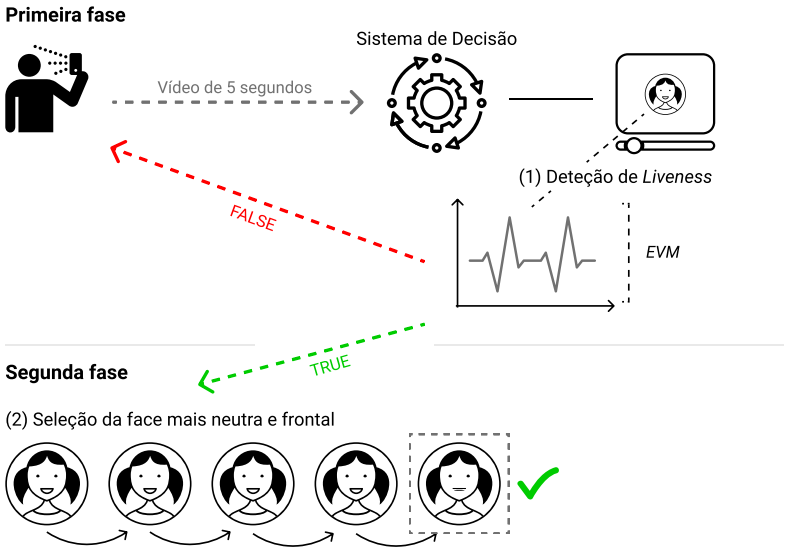


Figura 1.6: Decomposição do Sistema de Decisão.

1.4 Publicações

Desta dissertação foram submetidos os seguintes 3 artigos, e à data da submissão da dissertação, encontram-se em fase de revisão:

1. Ruslan Padnevych, David Semedo, David Carmo, João Magalhães. *"1-D Convolutional Neural Networks for Robust On the Fly Face Liveness Detection"* (Capítulo 3)

2. Ruslan Padnevych, David Semedo, David Carmo, João Magalhães. *"Robust Face Liveness Detection with Deep Convolutional Generative Adversarial Networks"* (Capítulo 4)

3. David Semedo, David Carmo, Ruslan Padnevych, João Magalhães. *"Contact-free Airport Borders with Biometrics-on-the-Move"*

Os dois primeiros artigos vêm a propósito de contribuir para o projeto de *SmartyFlow*, mais especificamente na implementação de um detetor de vivacidade, ou seja, detetor de ataques de apresentação.

O último artigo, é um artigo de demo, que representa toda a *framework* do sistema *SmartyFlow*, que está integrado no plataforma da *Vision-Box*, inclusivé as contribuições desta dissertação.

1.5 Estrutura do documento

Em relação à estrutura do restante documento, para facilitar a sua leitura, decidiu-se organizá-lo nos seguintes capítulos:

- O **Capítulo 2** refere-se ao **Trabalho relacionado**, que inclui um estudo acerca de Biometria, explica e exemplifica os diferentes procedimentos para efetuar *Presentation Attack Detection (PAD)* e, por fim, apresenta a análise de algoritmos já existentes que ajudam a determinar a existência dessas ameaças.
- O **Capítulo 3** apresenta 3 modelos baseados em **Redes Convolucionais 1D**, de diferentes complexidades, para deteção de vivacidade.
- No **Capítulo 4** é apresentado um modelo baseado em **Treino Adversarial** para geração de sinais de pulsação artificiais.
- O **Capítulo 5** contém a **Avaliação**, que começa por descrever o conjunto de dados utilizado e o processo da sua recolha. Seguidamente, são descritas algumas metodologias de avaliação aplicadas e, por fim, são apresentados e analisados os resultados das experiências realizadas.
- O **Capítulo 6** descreve as **Conclusões**, as aprendizagens e obstáculos enfrentados ao longo da elaboração desta dissertação. Como também, as **Futuras melhorias** da solução implementada, **Contribuições** e **Limitações**.

Capítulo 2

Trabalho relacionado

Este capítulo serve para analisar, discutir e perceber onde e como é que os trabalhos científicos existentes podem ser aplicados no contexto da presente tese.

Por isso, neste capítulo, a primeira secção surge com intuito de clarificar em que é que consiste a Biometria e a Biometria facial. Na segunda secção, são referidas as categorias em que os métodos de deteção de ataques de apresentação podem ser separados e alguns dos *International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 30107 standards* utilizados para definir *PAD*. Para terminar esta secção, são analisados possíveis procedimentos de deteção de ataques de apresentação. Por fim, as últimas secções apresentam algoritmos que podem ajudar a detetar os ataques de fraude de identidade e selecionar a face mais neutra que, posteriormente, é guardada numa Base de dados.

2.1 Biometria

O documento *ISO/IEC 2382-37:2017(E)* [40] estabelece uma descrição dos conceitos na área da biometria relativos ao reconhecimento de seres humanos e reconcilia os termos variantes que são utilizados nos padrões biométricos preexistentes com os termos preferenciais. Conforme o documento referido, a biometria é definida como sendo uma metodologia de reconhecimento automatizado de indivíduos com base nas suas características biológicas e comportamentais.

A biometria surge como uma substituição às palavras-passe, acrescentando também benefícios na autenticação de utilizadores. Esta substituição elimina a necessidade de ter que inventar, lembrar e alterar senhas complexas regularmente. Pois, se as credenciais de acesso são algo que fazem parte do próprio indivíduo, então estas nunca serão esquecidas. Quaisquer dados biométricos representam unicamente um indivíduo e devem permitir que este consiga efetuar a autenticação com segurança isto é, de forma que posteriormente ninguém tenha acesso aos seus dados. Esta tecnologia inovadora demonstra muitas promessas e pode mesmo revolucionar o acesso as informações confidenciais.

A biometria é uma metodologia bastante complexa que em certas situações supera as capacidades humanas. Como exemplo desta superação temos: os humanos têm uma excelente capacidade de identificar rostos conhecidos mas é muito complicado de lidar com um grande número de faces desconhecidas. Desta forma, essa limitação é ultrapassada pelos supercomputadores que utilizam vários algoritmos bastante sofisticados. Felizmente, com os avanços tecnológicos, criaram-se diversas alternativas para as biométricas convencionais que são mais seguras. Hoje em dia, é possível afirmar que o desempenho da tecnologia biométrica em ambientes controlados atingiu um nível satisfatório; no entanto, continuam a existir alguns pontos menos fortes que podem ser explorados em ambientes não controlados [34].

Os diferentes tipos biométricos são separados principalmente em dois grupos, baseados nas características fisiológicas e comportamentais de um indivíduo. Querendo isto dizer que, cada chave digital gerada é intrínseca ao seu respetivo utilizador, dificultando desta forma a sua replicação e partilha.

Atualmente, a autenticação através da verificação biométrica é cada vez mais utilizada para diversos fins, tais como controle de acesso nas fronteiras/empresas, identificação de suspeitos, autenticação de *smartphones*, serviços *online*, entre outros. Quando se fala em biométricas assume-se de imediato que se trata de impressão digital ou reconhecimento facial, mas existe uma grande variedade de tipos biométricos que são utilizados para facilitar a identificação e autenticação dos indivíduos.

Os seguintes diagramas 2.1 e 2.2 são, respetivamente, as representações dos dois conjuntos biométricos anteriormente referidos, seguidos de uma breve explicação de cada um dos tipos [52, 55].

- **Reconhecimento da Impressão digital** - permite que uma pessoa seja autenticada ou identificada através da análise e comparação das *ridges* e *valleys* dos seus dedos. Após capturar a impressão, os algoritmos sofisticados utilizam a imagem para produzir um modelo biométrico digital que identifica exclusivamente um indivíduo.

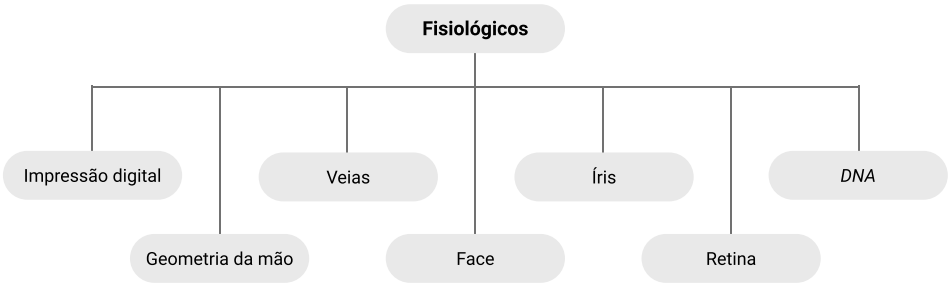


Figura 2.1: Diagrama com tipos de biometria fisiológicos.

Posteriormente, o modelo é comparado às impressões digitais novas ou já existentes para confirmar ou negar a correspondência. Este tipo biométrico foi uma das primeiras técnicas utilizadas para identificar automaticamente as pessoas.

- **Reconhecimento da geometria da mão** - segundo Sanchez-Reillo [36], é considerado um tipo biométrico que atinge um nível elevado de aceitação do utilizador com baixo custo computacional. Este tipo refere-se à medição das características da mão, como por exemplo o comprimento e a largura dos dedos, a sua curvatura e a posição relativa a outras características da mão para identificar um indivíduo.

- **Reconhecimento de veias** - este método é considerado como sendo o mais difícil de falsificar em comparação com os outros métodos, pois os padrões das veias no dedo ou na palma da mão estão localizados profundamente na pele. Com base nestes padrões, é efetuada a identificação de um indivíduo.

- **Reconhecimento facial** - utilizando a face do indivíduo é realizada uma comparação com todas as faces, que já se encontram presentes na Base de dados, para encontrar a face mais parecida. Conseguindo desta forma identificar o indivíduo. Visto que, a tese descrita neste documento recorre a utilização da biometria facial e sendo o reconhecimento facial um dos seus processos então, a sua explicação mais pormenorizada está descrita na seguinte secção 2.1.1.

- **Reconhecimento da íris** - a íris consiste em músculos grossos, semelhantes a fios. Os músculos, por sua vez, contêm dobras únicas que ao serem medidas, utilizando ferramentas de autenticação biométrica, podem confirmar a identidade com alta precisão.

- **Reconhecimento da retina** - é uma técnica que utiliza os padrões das veias unicamente existentes na retina para identificar uma pessoa.

- **Correspondência de *DNA*** - este processo é especialmente valioso, pois consegue lidar com casos tais como o desaparecimento de pessoas, identificação de vítimas em desastres e potencial tráfico de pessoas. O *DNA* coletado, por exemplo, do cabelo ou da saliva, contém *Short Tandem Repeat sequences (STRs)*. Os *STRs* do *DNA* podem confirmar a identidade de um indivíduo comparando-os com outros *STRs* existentes num banco de dados. Apesar de ser um processo muito preciso, é dificilmente aplicável num contexto como o do aeroporto, uma vez que, requer bastante tempo para a sua realização.

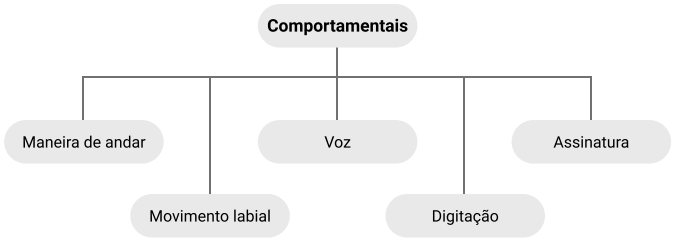


Figura 2.2: Diagrama com tipos de biometria comportamentais.

- **Maneira de andar** - esta biometria regista padrões no estilo da caminhada de um indivíduo através das imagens de vídeo. Posteriormente, estes padrões permitem transformar os dados mapeados em uma equação matemática. Este tipo de biometria é discreto, sendo ideal para vigilância massiva de multidões, pois consegue facilmente identificar as pessoas ao longe.

- **Movimento labial** - a autenticação biométrica do movimento labial rastreia e regista os movimentos musculares ao redor dos lábios para determinar se os mesmos seguem um padrão esperado. Os sensores biométricos de movimento labial geralmente exigem que os indivíduos verbalizem palavras-passe e registem o movimento labial correspondente para permitir ou negar o acesso.

- **Reconhecimento de voz** - para que um indivíduo seja corretamente identificado através da sua voz, é necessário realizar dois processos, identificação e verificação/autenticação. Na parte da identificação é criada uma lista de modelos que mais se assemelham com a voz procurada. Em seguida, é executada a verificação para determinar a correspondência conclusiva, ou seja, quando a voz do indivíduo corresponde exatamente a um dos modelos da lista.

- **Reconhecimento de digitação** - para estabelecer a identidade de um indivíduo é possível extrair características únicas da forma de digitação. As medidas podem incluir, por exemplo, o tempo necessário para pressionar cada tecla, o atraso entre dois pressionamentos consecutivos ou o número de caracteres digitados por minuto.

- **Reconhecimento da assinatura** - um tablet digital analisa e regista o estilo de caligrafia, em particular, da assinatura para criar automaticamente um perfil biométrico para as futuras autenticações.

Depois desta breve explicação, conclui-se que cada indivíduo constitui um perfil único, até mesmo os gémeos diferem no seu comportamento e composição física. A tecnologia biométrica diferencia características exclusivas para comprovar a identidade e melhorar a segurança.

2.1.1 Biometria Facial

A biometria facial é uma tecnologia recente que se tem tornado tendência na identificação, verificação ou autenticação de indivíduos em diversos setores. Este método biométrico é a base da tese descrita neste documento, pois é utilizado em conjunto com outros algoritmos para resolver o problema descrito no Capítulo 1.

Para contextualizar, a biometria facial pode ser aplicada em ambientes como: sistemas de saúde para reconhecer o cliente, qualquer empresa que necessite identificar o funcionário para fazer o controlo de acesso, aeroportos que é área em que esta tese se enquadra (Figura 1.4), entre outros.

O seu devido funcionamento apenas necessita de uma câmara e um *software* com capacidade de efetuar os seguintes três processos: **reconhecimento/identificação facial, verificação facial e autenticação facial**. O que é bastante positivo para as empresas, pois não é necessário investir em *hardware* específico de elevado custo. Além disso, é um processo que é difícil de falsificar mas que, infelizmente, hoje em dia, ainda não é impossível.

Antes de mencionar os desafios que esta técnica apresenta, é necessário esclarecer de uma forma clara a diferença entre os processos, mencionados acima, que são fundamentais na definição da biometria facial.

Reconhecimento/Identificação facial

Hoje em dia, existem diversas abordagens para efetuar o reconhecimento facial como por exemplo, *Appearance-Based*, *Feature-Based ou Soft Computing-Based* [34]. Para resolver o desafio em questão, utilizou-se a técnica *Video-Based* que permite estabelecer a identidade de uma ou várias pessoas presentes no vídeo com base nas suas características faciais. Dado o vídeo da face, uma abordagem típica do método de reconhecimento facial combina as características temporais[1] do movimento facial com as alterações de aparência para criar um padrão específico do indivíduo, contendo informações distintas que melhoram o desempenho [6]. O reconhecimento com base em vídeo é particularmente útil em cenários de vigilância nos quais talvez não seja possível capturar apenas uma boa imagem como é exigido pela maioria dos métodos baseados em imagens estáticas.

Esta tarefa procura identificar uma pessoa ou uma biometria desconhecida respondendo as perguntas como: "Quem é este indivíduo?" ou "Quem gerou esta biometria?". Para conseguir responder é necessário comparar a biometria apresentada com todas as outras que já estão armazenadas no banco de dados e encontrar a mais semelhante. É considerado como sendo uma correspondência de 1:N, onde o N é o número total de biometrias presentes na Base de dados [56].

No diagrama de caso de uso, Figura 1.4, apresentado no capítulo anterior, esta técnica de identificação é utilizada no ponto (4). Nessa situação:

1. A câmara deteta a face do indivíduo, representado na Figura 2.3 em (1).

2. Extraem-se as características únicas que identificam cada sujeito, situação (2) na Figura 2.3.

3. Realiza-se uma pesquisa (1:N) na Base de dados, ponto (3) da Figura 2.3, a procura da face que mais se assemelha.

4. Identifica-se a face que representa a melhor correspondência em termos das características extraídas anteriormente, Figura 2.3 referência (4).

[1] A caracterização temporal da face é o processo de modelação da forma como a aparência da face varia com o tempo. À medida que a face se move ao longo do vídeo, a sua aparência altera devido as alterações na pose, expressão, condições de iluminação, etc [26].

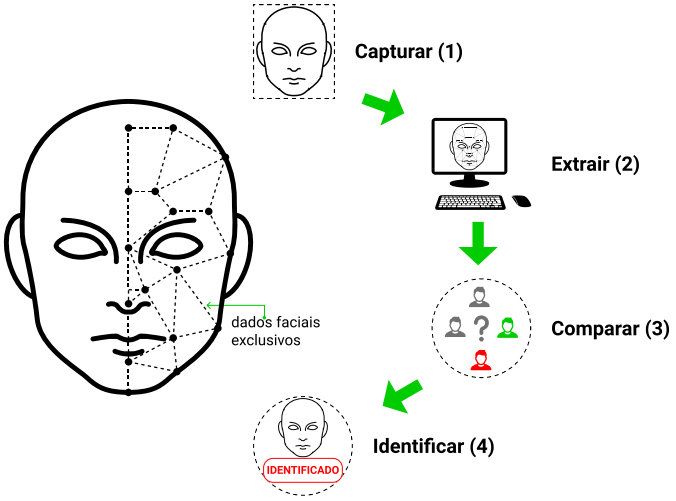


Figura 2.3: Funcionamento do processo de Reconhecimento/Identificação facial.

Verificação facial

A verificação facial é um processo um pouco diferente em comparação ao reconhecimento/identificação facial. Esta tarefa procura certificar-se de que a presente pessoa é mesmo quem ela garante ser, ou seja, respondendo à pergunta "É mesmo este indivíduo?". Para isso, verifica se a biometria apresentada corresponde exatamente a um perfil biométrico associado a este indivíduo que já se encontra armazenado no banco de dados. Ao contrário da tarefa descrita em 2.1.1, esta é considerada como sendo uma correspondência de 1:1 pois, o processo tenta corresponder a biometria apresentada pelo indivíduo com uma biométrica específica já registada.

Visto que apenas é necessário efetuar uma comparação de 1:1, os resultados são obtidos muito mais rapidamente e com mais precisão em comparação a tarefa de identificação facial [56].

Autenticação facial

Este processo ajuda a decidir se o indivíduo apresentado pode proceder com a sua intervenção. Para tal, verifica se este tem acesso/permissão para continuar com a ação pretendida ou se se trata de um *Presentation Attack (PA)*. Este tipo de ataques podem ocorrer durante a autenticação, se e só se, o artefacto[2] ou também designado de *Presentation Attack Instrument (PAI)*[3] apresentado é de um indivíduo legítimo que está registado

[2]Objeto ou representação artificial que apresenta cópia/imitação das características biométricas ou padrões biométricos sintéticos [39].

[3]Uma característica ou um objeto biométrico utilizado num ataque de apresentação [39].

no sistema biométrico, ou seja, já se encontra guardado na Base de dados [33]. Um dos ataques mais comuns devido à alta exposição do rosto e ao baixo custo de reprodução, é o impostor[4] apresentar uma fotografia ou um vídeo de um indivíduo verdadeiro para o sensor, a fim de ser falsamente autenticado pelo sistema [19, 44].

Outros desafios que podem ocorrer na biometria facial estão descritos com mais detalhes na próxima secção 2.1.1.1.

2.1.1.1 Desafios

Cada dado biométrico pertence somente a um indivíduo; Portanto apenas este deve poder utilizá-lo. No entanto, quando uma biometria é usada para proteger algo valioso como por exemplo, a identidade de um indivíduo, os ditos impostores tentam enganar o sistema com falsificações biométricas [50]. Por isso, os sistemas têm de estar preparados para lidar com os desafios como por exemplo:

- A iluminação do ambiente em que o indivíduo se encontra, pois estando mais escuro ou mais claro pode dificultar a sua identificação.
- O indivíduo pode-se apresentar com óculos, barba, maquilhagem ou até mesmo com diferentes tipos de lentes. Logo, para o sistema não será tão fácil encontrar a correspondência, uma vez que, o aspeto da pessoa terá traços distintos em comparação aos que estão guardados na Base de dados.
- Ocorrência de ataques de apresentação, isto é, passar-se por outro indivíduo que podem ser separados em duas categorias, tal como é possível observar na Tabela abaixo [38].

Tabela 2.1: Alguns dos ataques de apresentação.

	Estáticos	**Dinâmicos**
2D	Fotografia, máscara lisa de papel/plástico	Reprodução de vídeo no ecrã, várias fotografias mostradas uma a uma
3D	Impressão 3D, escultura, máscara	Robôs que reproduzem expressões, maquilhagem bem preparada

É importante referir que a probabilidade de um ataque ocorrer com sucesso varia consideravelmente, dependendo muito das características do sistema de biometria facial,

[4]Sujeito que tenta corresponder à referência biométrica da outra pessoa [40].

como por exemplo, se utiliza a luz visível ou outra faixa do espectro, se tem apenas um sensor ou tem vários, a resolução. Por outro lado, as características do artefacto apresentado também são bastante relevantes tais como, qualidade da textura, aparência, resolução do dispositivo, entre outros [19].

No contexto desta dissertação, o registo no sistema pode ser efetuado num ambiente não controlado, por isso, o sistema está exposto a diferentes tipos de ataques. Os ataques que vão ser considerados como base dos testes aos algoritmos a desenvolver são: diferentes tipos de iluminação e os ataques de 2D estáticos referidos na Tabela 2.1. Com estas simulações pretende-se testar a segurança dos algoritmos implementados de *Face anti-spoofing detection*.

Deste modo, devido à existência dessas ameaças, as organizações antes de introduzir a biometria facial nos seus sistemas, têm de garantir que possuem um sistema de anti-falsificação capaz de proteger dados sensíveis, reduzir roubos e mitigar fraudes. Para que, as tarefas de identificação, verificação e autenticação dos funcionários sejam efetuadas com precisão e segurança.

2.2 Deteção de ataques de apresentação

Visto que, ultimamente a biometria facial tem sido implementada em vários setores, é necessário assegurar que esta seja resiliente a qualquer artefacto. Por isso, têm sido desenvolvidos inúmeros métodos de determinação automática de ataques de apresentação, que são denominados como *PAD*. Todas as técnicas que são capazes de distinguir de forma autónoma entre características biométricas reais e artefactos produzidos sinteticamente podem ser considerados como métodos de *PAD*. A biometria facial sem a utilização dessas técnicas está claramente exposta até a ameaças mais simples que uma pessoa comum conseguiria facilmente identificar. No entanto, sempre que um novo ataque é descoberto é necessário desenvolver e adaptar novas contramedidas [19].

Para detetar se as características biométricas, que são apresentadas no sensor, são realmente originadas por um indivíduo verdadeiro pode ser realizado de quatro formas distintas [2]:

1. Com sensores, intrusivos, capazes de descobrir no sinal características padrão com indícios de vivacidade.

2. Com *hardware* auxiliar, não intrusivo, dedicado para detetar evidências de vivacidade, que nem sempre é possível implantar devido por exemplo ao elevado custo.

3. Com um método de *challenge-response*, solicitando que o utilizador interaja com o sistema de uma maneira específica.

4. Com *hardware* não intrusivo e não especializado, como por exemplo uma câmara simples *RGB*. Para isso, é necessário investir no desenvolvimento de algoritmos que sejam extremamente robustos contra ataques.

Por isso, o âmbito desta tese enquadra-se no ponto 4., acima referido, uma vez que, é necessário desenvolver um método de *PAD*, mais concretamente *Face anti-spoofing detection*, cujo o fator de decisão é deteção da vivacidade. Uma das formas de detetar a vivacidade é através do batimento cardíaco.

O batimento cardíaco é uma das características fisiológicas mais importantes do corpo humano. Uma vez que, contém informações vitais sobre o sistema cardíaco e arritmia. As informações obtidas através da pulsação são úteis não apenas para fins médicos, mas também em outras áreas, como reconhecimento de expressões ou deteção da vivacidade. Esta característica apresenta vantagens como: (i) difícil de falsificar; (ii) contém bastante informação sobre um indivíduo; (iii) pode ser medida sem o auxílio de um *hardware* especializado. Normalmente, o batimento cardíaco é obtido com equipamento especializado e de forma intrusiva. Em alternativa, é possível obter uma estimativa do batimento através do vídeo recolhido da face do indivíduo. Uma das técnicas conhecidas para este processo é o *EVM* que com base nas micro flutuações da cor na face estima o batimento cardíaco. Este processo não é intrusivo e pode ser realizado com uma simples câmara *RGB*. No entanto, o sinal do batimento obtido através desta técnica está sujeito a vários tipos de ruído, o que faz com que seja necessário desenvolver um algoritmo que consiga isolar os vários tipos de ruído de forma a obter um sistema robusto de deteção da vivacidade.

As abordagens de *PAD* podem ser decompostas em dois grupos [19, 33]: ***Hardware-based*** e ***Software-base*** como é possível visualizar no esquema da Figura 2.4.

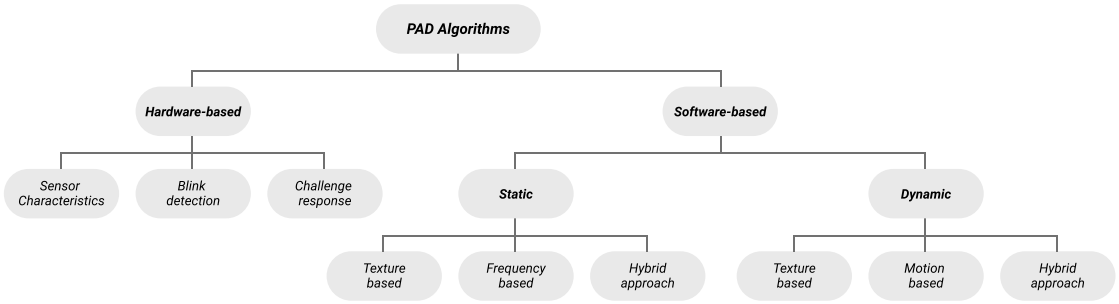


Figura 2.4: Classificação dos algoritmos de deteção de ataques de apresentação.

2.2.1 *Hardware-based*

Este tipo de abordagens procura explorar as características do rosto do indivíduo com ajuda de componentes adicionais de *hardware* que funcionam em conjunto com os sensores de reconhecimento facial. Nas abordagens deste tipo, podem ser utilizados sensores que localizam a face ou ser exigido que haja uma interação direta com o equipamento. Apesar de estes métodos conseguirem bons resultados na deteção de ataques biométricos, introduzem também sobrecarga computacional e custos adicionais de pesquisa. Cada uma das abordagens podem ser classificadas num dos três tipos: ***Sensor characteristics***, ***Blink detection*** ou ***Challenge-response***.

- ***Sensor characteristics*** - baseado na exploração das características da câmara (ou sensor) utilizada para capturar a imagem da face (ou vídeo). As características dependem do tipo do sensor utilizado, por exemplo, medir a variação do foco com *light field camera (LFC)* ou refletância dos sensores *near-infrared /thermal /multispectral* [33].

- ***Blink detection*** - uma medida bastante utilizada para deteção de vivacidade para mitigar os ataques de apresentação. A ideia por trás desta abordagem é seguir continuamente o piscar dos olhos que acontece de forma inconsciente. Esta abordagem consegue prevenir eficientemente contra ameaças cujo artefacto é uma fotografia. No entanto, continua a ser vulnerável a ataques de vídeo. O método de *Blinking detection* tanto pode ser implementado utilizando equipamento dedicado ou técnicas baseadas em *software* [33].

- ***Challenge-response*** - fornece uma interface de utilizador com objetivo de gravar as respostas aos desafios que vão sendo propostos ao indivíduo, como por exemplo, olhar numa certa direção. Posteriormente, as respostas são processadas para se certificar que a apresentação é genuína. Com esta abordagem é possível identificar os ataques que sejam estáticos, como por exemplo, uma fotografia.

2.2.2 *Software-based*

As abordagens baseadas em *software* envolvem um algoritmo que pode determinar se a face capturada decorre de um ataque de apresentação ou é uma apresentação *bona fide*, isto é, genuína. Por sua vez, estas abordagens podem ser divididas em duas categorias, dependendo se têm ou não em consideração as informações temporais [19]: ***Static*** e ***Dynamic analysis***.

2.2.2.1 *Static analysis*

Análises estáticas, tal como o nome sugere, são métodos que apenas trabalham com uma imagem, sem a necessidade da informação temporal. No entanto, podem ser aplicados também a uma sequência de *frames*, em que a decisão final é tomada com base nos resultados dos *frames* que ocorrem mais vezes [33]. Visto que, são utilizadas apenas para uma imagem de cada vez não é necessário nenhuma tecnologia especial para a sua realização. Por isso, têm baixo custo de implementação, conseguem obter um bom desempenho e baixo tempo de execução o que as torna mais rápidas do que *Dynamic analysis*. As abordagens estáticas podem ser separadas em três grupos, dependendo da natureza dos algoritmos: ***Texture-based***, ***Frequency-based*** e ***Hybrid***.

- ***Texture-based approach*** - a ideia principal por trás desta abordagem é aprender e detetar a estrutura das micro texturas que caracterizam as faces reais. Ao criar falsificações, normalmente são introduzidas degradações de qualidade, possibilitando distinguir entre uma tentativa de acesso genuína e um ataque, analisando as suas texturas [12]. Além da textura, existem outras propriedades da face e da pele humana que podem ser exploradas para diferenciar as amostras tais como, absorção, reflexão, refração e dispersão. Esta forma de análise tem sido efetivamente usada na deteção de ataques cujo artefacto é uma fotografia, vídeo ou máscara, uma vez que, todos estes apresentam texturas diferentes em comparação ao rosto real [19].

- ***Frequency-based approach*** - esta abordagem funciona de forma semelhante à abordagem baseada na textura, aproveitando o facto de a pele ter um componente de frequência mais alto [35] numa imagem ao vivo do que numa fotografia impressa ou em exibição.

- ***Hybrid approach*** - consiste na combinação de diferentes métodos que envolvem textura, forma, frequência, entre outros.

2.2.2.2 *Dynamic analysis*

Este tipo de métodos têm como objetivo detetar os ataques de apresentação com base na análise da informação temporal do vídeo apresentado. As abordagens dinâmicas tendem a modelar essas informações temporais ao explorar a existência de movimentos ao longo dos *frames*. As análises podem consistir na deteção de qualquer sinal fisiológico, como por exemplo, piscar dos olhos, alterações na expressão facial ou movimentos da

boca [19]. Tal como é possível observar na Figura 2.4, estas abordagens podem ser separadas nos seguintes grupos: ***Texture-based***, ***Motion-based*** e ***Hybrid***.

- ***Texture-based approach*** - esta técnica funciona de forma semelhante à abordagem estática apenas com uma pequena variante, é aplicada a cada *frame* do vídeo apresentado. Desta forma, esta abordagem apresenta maior carga computacional em comparação à abordagem estática.
- ***Motion-based approach*** - captura os movimentos subconscientes provocados pelos músculos da face. Os movimentos são particularmente criados pelos movimentos da cabeça, boca ou olhos.
- ***Hybrid approach*** - explora uma combinação das duas abordagens anteriores, *Texture-based* e *Motion-based*, para obter um desempenho preciso na identificação de *video replay attacks* nos sistemas de reconhecimento facial [33].

2.2.3 *ISO/IEC 30107 standards* de *Presentation Attack Detection*

O objetivo do *ISO/IEC 30107 standards* é providenciar uma base para os métodos de deteção de ataques de apresentação através da definição dos termos e estabelecimento de uma estrutura que permita que os eventos de ataque de apresentação possam ser especificados, detetados, categorizados, detalhados e comunicados a outros subsistemas biométricos. As normas presentes no documento anteriormente referido não defendem nenhum método padrão específico de *PAD*, pois o âmbito é limitado apenas à descrição dos ataques que ocorrem ao nível do sensor biométrico durante a apresentação e recolha das características biométricas. Quaisquer outras ameaças são consideradas fora do escopo do *ISO/IEC 30107* [39].

Os ataques de apresentação podem surgir com diferentes intenções, ou seja, provenientes de diferentes tipos de atacantes, aqueles que pretendem passar-se por outro indivíduo (*impostor*) ou aqueles que não querem ser reconhecidos (*identity concealer*[5]).

Os impostores biométricos podem realizar os ataques de duas formas distintas. No primeiro subtipo, o atacante pretende ser reconhecido como sendo um indivíduo específico conhecido pelo sistema. No segundo subtipo, pretende apenas ser reconhecido não especificando exatamente o indivíduo. Por outro lado, *biometric concealer* tenta ao máximo proteger as suas características biométricas, por isso, recorre à criação de características ou objetos falsificados (*spoofs* ou *PAI*) que representam outro indivíduo: artefactos, mutilações e repetição.

[5]Sujeito que tenta evitar corresponder à sua própria referência biométrica [40].

Por vezes, os sistemas biométricos ou os métodos de *PAD* não conseguem diferenciar entre os ataques de apresentação cujo objetivo é interferir na operação do sistema e as apresentações não conformes. Assim sendo, é necessário garantir que qualquer dúvida que surja seja rejeitada pelos métodos de deteção de ataques de apresentação pois, nesta área é fundamental garantir a total segurança dos cidadãos.

De acordo com as normas definidas no standard *ISO/IEC 30107* [39], o processo de deteção de ataques de apresentação é idêntico ao processo de reconhecimento biométrico e pode ser realizado com os seguintes passos:

1. Recolher os dados biométricos do indivíduo para serem processados pelo método de *PAD*.
2. Extrair as características dos dados recebidos.
3. Comparar as características extraídas com os critérios.
4. Resultado da comparação (deteção, não deteção, pontuação, entre outros). A decisão final é tomada com base no resultado da comparação.

2.2.4 Exemplos de métodos de deteção de ataques de apresentação

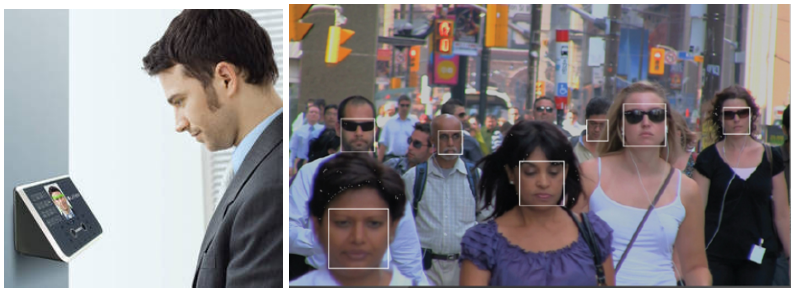
Os métodos de deteção da vivacidade são um subconjunto dos métodos de deteção de ataques de apresentação ao nível do subsistema de recolha de dados. Uma vez que, o propósito desta tese é desenvolver algoritmos cujo objetivo é detetar a existência da vivacidade ao nível do sensor, a seguir são descritas técnicas que ajudam a mitigar os ataques do tipo falsificação do rosto recorrendo a deteção de vivacidade.

Abordagens ativas e passivas

Para detetar se o indivíduo apresentado tem propriedades de vivacidade pode-se recorrer a dois tipos de abordagens: ativas e passivas.

As abordagens ativas, tal como o nome sugere, requerem algum tipo específico de atividade/ação por parte do utilizador ao interagir diretamente com o sistema (Figura 2.5a). O Movimento labial e *Challenge-response* são apenas dois dos exemplos das atividade relacionadas com abordagens ativas que se encontram mais detalhadas nas secções 2.1 e 2.2.1 respetivamente.

Contrariamente, as abordagens passivas (Figura 2.5b), ou também chamadas de observações não estimuladas de vivacidade, não exigem uma interação direta com o sistema biométrico. Neste caso, a vivacidade é caracterizada exclusivamente pelo que é



(a) [53]. (b) [49].

Figura 2.5: (a) Técnica ativa e (b) passiva de deteção da vivacidade.

recebido através do sensor durante um período de tempo apropriado, sem estímulos intencionais relacionados à vivacidade [39]. As medições que podem ser efetuadas nestas abordagens, utilizando a face de um indivíduo são, por exemplo, pulsação e absorção da frequência da luz pelo sangue/pele.

É importante referir que a pulsação tanto pode ser medida utilizando abordagens ativas ou passivas. As abordagens ativas (intrusivas) podem exigir por exemplo que o indivíduo coloque uma pulseira no pulso para medir a pulsação. Por outro lado, as abordagens passivas podem efetuar a medição da pulsação utilizando *EVM framework* como está explicado mais adiante na secção 2.2.4.

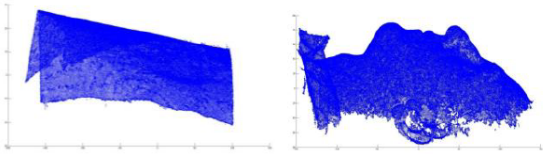
Visto que o problema desta tese envolve a pulsação, é necessário esclarecer qual das abordagens seria mais apropriada para ser utilizada nos aeroportos. As abordagens intrusivas apesar de conseguirem medir a pulsação com mais precisão, impedem uma progressão contínua no controlo de fronteiras e são mais dispendiosas. Por isso, no ambiente em questão, as abordagens passivas são mais apropriadas uma vez que são mais cómodas e apresentam um custo baixo de implementação.

Imagem facial 3D

Este método calcula as características tridimensionais dos dados presentes no rosto 2.6b para determinar se a face capturada pelo sensor é real ou não, isto é, se existem evidências de vivacidade ou não 2.6a.

Observando os dois gráficos presentes na Figura 2.6b, consegue-se facilmente concluir que o que está à esquerda é uma representação de um ataque utilizando a fotografia do indivíduo, pois não representa quaisquer características tridimensionais do rosto. Por outro lado, o gráfico da direita representa a forma de uma face 3D de um indivíduo o que permite concluir que é mesmo um indivíduo real.

(a) Exemplo de ataque de apresentação ao nível do sensor [25].



(b) Extração das características 3D de uma imagem da face do indivíduo (esquerda), e de uma face 3D real (direita) [25].

Figura 2.6: Deteção da vivacidade utilizando a técnica da imagem facial 3D.

Segundo Lagorio et al. [25], existem diversas vantagens na utilização desta abordagem em comparação com as outras técnicas de deteção de vivacidade:

- Não requer interação direta com o indivíduo (sorrir, falar ou responder a qualquer solicitação).
- Não é necessário nenhum tipo de *hardware* adicional, como por exemplo, microfone, pois o sistema de obtenção das características 3D utilizado na fase de reconhecimento pode ser, simplesmente, adaptado numa etapa de pré-processamento. Se são utilizadas faces bidimensionais para efetuar o reconhecimento, um conjunto dessas imagens, obtidas de vários ângulos, pode ser utilizado para reconstruir a forma tridimensional do rosto.
- Não existem quaisquer restrições na pose e na orientação da cabeça.

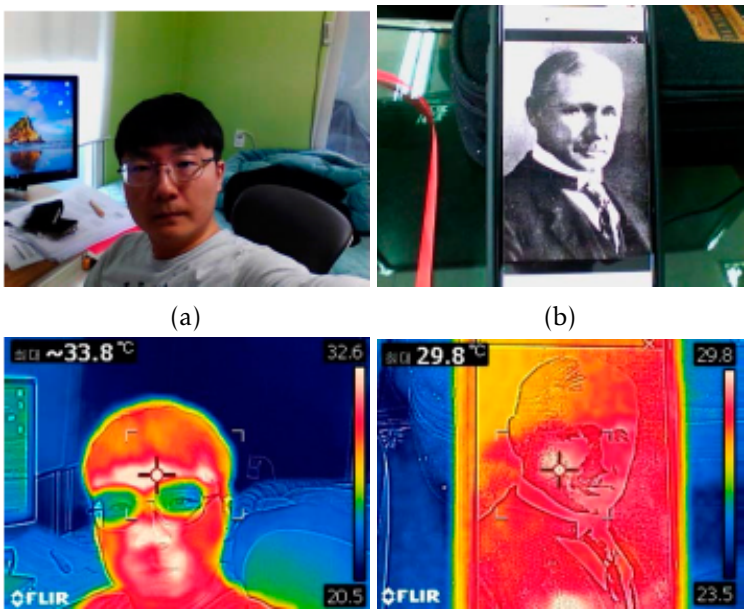
Atualmente, existem vários algoritmos de reconhecimento facial 3D com níveis de correspondência muito elevados que permitem atenuar ataques de falsificação do rosto. No entanto, a tecnologia tridimensional continua a não ser amplamente utilizada em aplicações práticas, devido ao facto de requerer um custo elevado, uma maior complexidade computacional e um pré-tratamento de imagens complicado. Sendo por isso, é ineficiente a aplicação deste método num ambiente como o do aeroporto, pois o que se pretende é uma tecnologia rápida para poder evitar as filas no controlo de fronteiras.

Temperatura facial

Devido à crescente procura de melhores sistemas de segurança e de melhor vigilância nas áreas noturnas e com pouca iluminação, as câmaras térmicas têm sido incluídas

em alguns sistemas. As câmaras de infravermelho (IV) térmico têm a capacidade de capturar imagens e objetos com base na refletância da luz IV ou na emissão de radiação IV [24]. A radiação infravermelha é uma radiação eletromagnética emitida proporcionalmente ao calor gerado/refletido por um objeto/indivíduo e, portanto, a imagem IV é chamada imagem térmica ou termograma (Figura 2.7c e 2.7d). Esta abordagem de identificação é não intrusiva uma vez que, a câmara pode capturar a face do indivíduo a uma determinada distância.

Citando a autora Sund Levander et al. [41], a temperatura média de um corpo real e vivo varia entre os 36 e 37 graus. Por isso, utilizando a câmara térmica consegue-se obter uma temperatura aproximada do indivíduo.



(a) (b)

(c) (d)

Figura 2.7: Dados exemplo: (a) rosto real tirado com uma lente *RGB*; (b) rosto no visor de um dispositivo tirado com uma lente *RGB*; (c) rosto real tirado com uma lente infravermelha; (d) face no visor de um dispositivo tirada com uma lente infravermelha [37].

Caso o valor esteja bastante afastado do intervalo estabelecido, (Figura 2.7d, temperatura 29.8 graus) conclui-se então a existência de um ataque. Caso contrário (Figura 2.7c, temperatura 33.8), considera-se que é um indivíduo genuíno e que de facto existem indícios de vivacidade.

Esta abordagem é pouco utilizada nos projetos de grande escala, visto que, as câmaras térmicas são bastante dispendiosas. Sendo por isso, o objetivo da tese descrita neste documento é detetar a vivacidade utilizando uma câmara simples *RGB* para reduzir os

custos de implementação.

Eulerian Video Magnification

As imagens e os vídeos contêm informações valiosas sobre as mudanças que ocorrem ao longo de um período de tempo. Como declarado pelo autor Abbas et al. [1], os vídeos possuem imensa informação, que não é possível visualizar a olho nu. Por isso, é necessário ampliar essas pequenas alterações, utilizando as variações temporais, para estudar o seu comportamento. Por exemplo, na pele humana pode ocorrer ligeira variação da cor, devido ao fluxo sanguíneo, que não é possível observar a olho nu [1]. Essa pequena variação, designada de *Remote Photoplethysmography signal (rPPG)*[6], pode ser usada para estimar a pulsação humana sem qualquer contacto físico. Segundo os autores Liu et al. [28], o método utilizado para extrair o sinal *rPPG* têm duas principais desvantagens: (1) é sensível a variação de posicionamento e expressão, pois fica mais difícil de seguir uma área específica da face para medir as mudanças de intensidade da cor; (2) é sensível também às mudanças de iluminação, pois a iluminação extra afeta a quantidade de luz refletida na pele.

O algoritmo proposto por Abbas et al. [1] é baseado nos valores das cores de uma série temporal em qualquer localização espacial (pixel) e amplifica a variação de uma determinada faixa de frequência que seja de interesse. Uma faixa de frequência que contém informações vitais sobre a frequência cardíaca é automaticamente selecionada e amplificada.

Figura 2.8: Pulsação obtida com *EVM* [29].

Para estimar o batimento cardíaco a partir da face do indivíduo, amplificam-se as faixas de frequência que representam pulsações humanas plausíveis, revelando a variação da vermelhidão à medida que o sangue flui (Figura 2.8).

Ao obter a frequência cardíaca não se pode afirmar com toda a certeza que esta foi originada a partir de uma pessoa real, pois é possível simular um batimento cardíaco utilizando ataques de apresentação mais especificamente *face-spoofs*. Com esta dissertação pretende-se solucionar exatamente este problema. Utilizando o sinal *rPPG*, obtido

[6]Consiste na observação indirecta das variações do volume sanguíneo ao medir a absorção e a reflexão da luz na pele de um indivíduo [4].

através de *EVM*, é possível estimar a pulsação e com base nessa estimativa decidir se esta pertence a um indivíduo real, ou seja, se contém sinais de vivacidade ou se se trata de um impostor.

2.3 Algoritmos de deteção de ataques de apresentação

Com esta secção pretende-se analisar algoritmos que são capazes de detetar a presença de ataques de apresentação através da deteção da vivacidade, ou seja, verificar se um indivíduo está realmente presente ou se é um artefacto. No âmbito desta tese, este tipo de algoritmos irão compor o **Sistema de Decisão** presente na Figura 1.6. Para que o sistema seja confiável em cenários não supervisionados é necessário garantir que este seja *spoof-proof*.

2.3.1 Métodos *Face anti-spoofing*

As biometrias, como face e impressão digital, são amplamente utilizadas para autenticação de pessoas, devido à sua distinção intrínseca e conveniência de uso. A biometria facial é uma das modalidades mais populares que tem recebido crescente atenção nos últimos anos. No entanto, a atenção também traz um incentivo crescente para os impostores projetarem ataques de apresentação, para serem autenticados como sendo utilizadores genuínos. Devido à facilidade em obter um rosto humano, a falsificação facial pode ser tão simples como utilização de uma fotografia impressa (ataque de impressão) ou um vídeo (*replay attack*). Em algumas situações, estas falsificações podem estar visualmente muito próximas do rosto real do indivíduo genuíno, por isso, existe a necessidade em desenvolver algoritmos robustos de *Face anti-spoofing*. De seguida, serão analisados métodos propostos que procuram oferecer resiliência a ataques de *Face spoofing*.

Long-term Statistical Spectral

Esta abordagem foi primeiramente utilizada no contexto de deteção de ataque de voz, tendo conseguido com sucesso distinguir um orador real de uma gravação numa tarefa de autenticação através de voz. A principal vantagem das características de *LTSS* é a sua capacidade de lidar com qualquer tipo de sinal sem ser necessariamente sinal sonoro. Por isso, os autores Heusch e Marcel [20] sugerem a utilização dessas características no sinal de pulsação, que é estimado através do *rPPG* obtido pelo *EVM*, para discriminar acessos genuínos de tentativas de ataques.

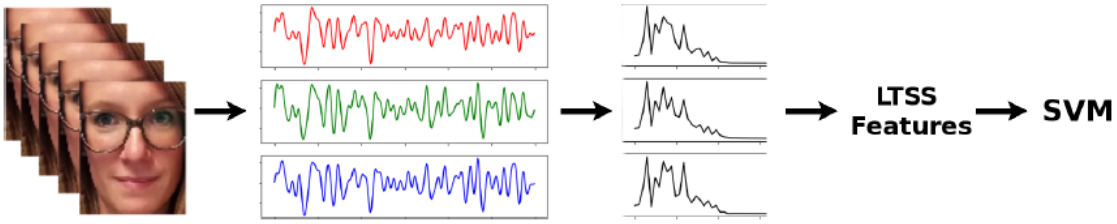


Figura 2.9: Visão geral da abordagem *LTSS* para detetar ataques de apresentação com base na pulsação [20].

As características *LTSS* são extraídas ao processar o sinal *rPPG* utilizando janelas temporais sobrepostas [20]. Segundo Heusch e Marcel [20], a análise temporal do conteúdo da frequência baseada em janelas temporais é adequada para os sinais de pulsação, pois os sinais de pulsação das tentativas reais devem conter alguma periodicidade, enquanto os sinais de pulsação de ataques não. Em cada janela deslizante w, o sinal *rPPG* é convertido de um domínio temporal para um domínio da frequência utilizando *Discrete Fourier Transform (DFT)* de N pontos [27]. Como resultado, é produzido um vetor X_w de coeficientes *DFT* de dimensão $k = \{0,\ldots,\frac{N}{2}-1\}$. Utilizando o conjunto de vetores de coeficientes *DFT* $X_1, X_2, \ldots, X_w$, as estatísticas de primeira e segunda ordem dos componentes de frequência são calculadas para o $k = \{0,\ldots,\frac{N}{2}-1\}$ [20] utilizando as fórmulas de média e variância.

Seguidamente, os vetores de média e variância são concatenados para representar as *spectral statistics* do dado sinal. Em consequência, é formado um vetor com uma característica que é *LTSS*. Por fim, os vetores com característica *LTSS* são utilizados em conjunto com *Support Vector Machine (SVM)*, como está representado na Figura 2.9, para classificar o dado vídeo como sendo um *bonafide* ou um ataque de apresentação.

Dynamic Mode Decomposition

O algoritmo *DMD* é uma abordagem totalmente *data-driven* (análise de movimento e análise de textura) para detetar os sinais de vivacidade. Este algoritmo tem uma propriedade exclusiva que é capacidade de representar as informações temporais de todo o vídeo como sendo uma única imagem. Conforme proposto por Tirunagari et al. [42], o processo de classificação representado na Figura 2.10 é composto por *DMD*, *Local Binary Patterns (LBP)* e *SVM* com o *histogram intersection kernel*.

Em primeiro lugar, o vídeo recebido como *input* é processado pelo algoritmo *DMD* para gerar as *dynamic mode images*. Das imagens geradas selecionamos apenas uma em que o *phase angle* é = 0 ou muito próximo desse valor [42]. Em segundo lugar,

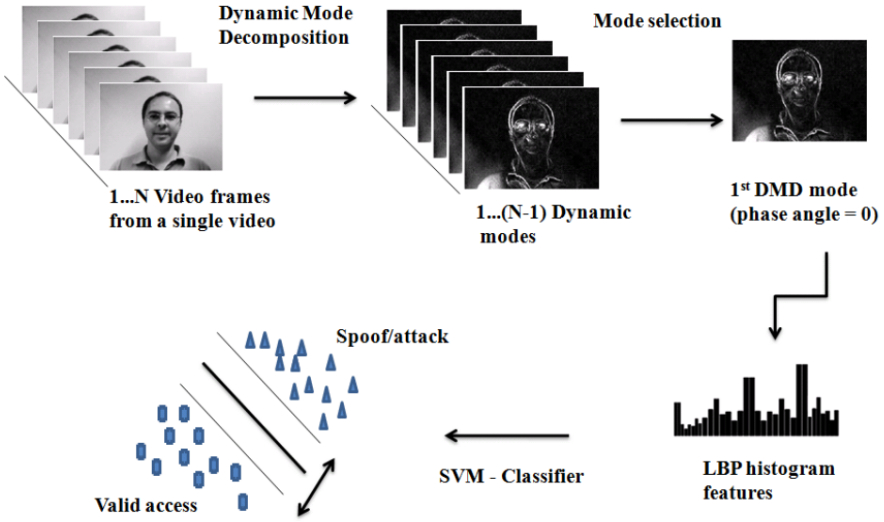


Figura 2.10: Visão geral da abordagem que envolve o algoritmo *DMD*.

LBP histogram features são calculadas para a imagem anteriormente selecionada. Para terminar, o código produzido por *LBP* é utilizado como *input* do classificador *SVM* previamente treinado para classificar se o vídeo representa um *bonafide* ou um ataque de apresentação. Como declarado pelo autor Tirunagari et al. [42], o *Half Total Error Rate* é utilizado para avaliar o desempenho desta abordagem.

2.3.2 *Convolution Neural Network* para detecção de vivacidade

CNN [14] é uma abordagem que pode ajudar a mitigar os ataques de apresentação classificando as imagens ou os vídeos da face do indivíduo como sendo reais, isto é, existência da vivacidade ou falsos.

Patch and Depth-Based Convolutional Neural Networks

A abordagem proposta por Yousef et al. [47] consiste em dois fluxos de *CNN*, como está representado na Figura 2.11.

Um deles extrai as características de aparência/textura e o outro calcula as *holistic depth maps* a partir de uma imagem do rosto. As características de aparência facilitam a *CNN* a discriminar as falsificações, utilizando áreas aleatórias (*random patches*) da face. Por outro lado, *holistic depth map* examina a face toda e verifica se existe profundidade na imagem da face recebida como *input*. Esta abordagem ajuda a resolver principalmente os ataques de impressão e os *replay attacks* [47].

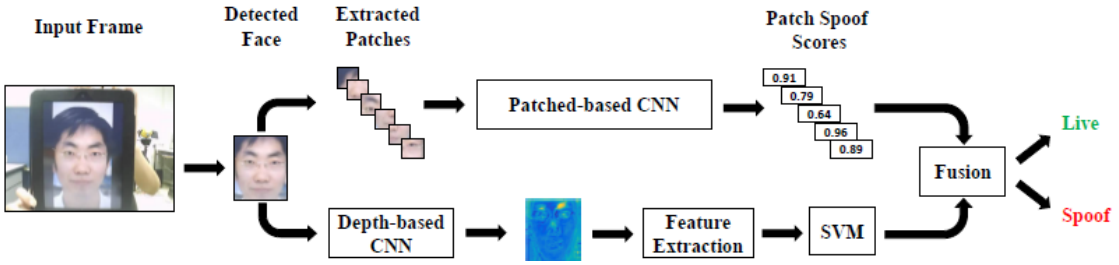


Figura 2.11: Arquitetura da abordagem *Patch and Depth-Based CNNs* proposta por Yousef et al. [47].

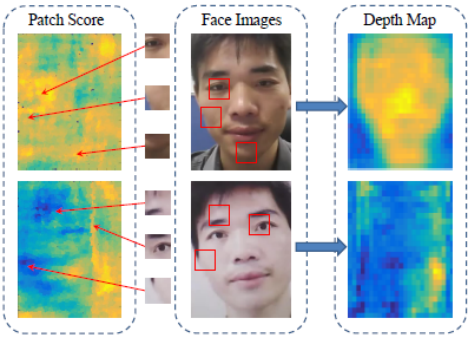


Figura 2.12: Visão geral da abordagem *Patch and Depth-based CNNs*. A coluna da esquerda representa as pontuações do **primeiro *CNN***, para a imagem real (em cima) e a imagem falsificada (em baixo). As cores azul/amarelo representam alta/baixa probabilidade de ser uma falsificação. A coluna da direita ilustra o resultado do **segundo *CNN***, em que as cores amarelo/azul representam os pontos próximos/afastados [47].

O **primeiro *Patch-based CNN***, utilizado para estudar as características de aparência (à esquerda na Figura 2.12), é *end-to-end trained* e atribui uma pontuação a cada *patch* extraída do rosto. A pontuação da imagem da face é obtida através da média das pontuações.

Relativamente ao **segundo *Depth-based CNN***, é treinada a *Fully Convolutional Neural Network (FCNN)* para estimar o *depth map* da imagem da face e atribuir-lhe uma pontuação de vitalidade com base no *depth map* estimado. De acordo com Yousef et al. [47], um ataque de apresentação de impressão ou reprodução tem um *depth map* plano, enquanto as faces reais têm uma *depth* da face normal (à direita na Figura 2.12).

No fim, a fusão das pontuações de ambas as *CNNs* leva à classificação final, *live* ou *spoof*. Uma imagem ou um vídeo de um rosto é considerado como falsificação se a sua pontuação estiver acima de um *threshold* predefinido.

Cada uma das *CNNs* referidas pode ser utilizada de forma independente para detetar

os ataques de apresentação. No entanto, os autores Yousef et al. [47] afirmam que a fusão de ambas fornece resultados promissores.

Esta abordagem pode ser aplicada no âmbito desta tese como sendo uma alternativa aos métodos de deteção de ataques de apresentação através do batimento cardíaco.

2.4 Algoritmo de seleção da face mais neutra e frontal

O objetivo desta secção é analisar algoritmos que permitem, a partir de um vídeo, selecionar a *frame* que contém a face mais neutra e frontal de um sujeito (como está ilustrado na **Segunda fase** da Figura 1.6). A melhor face é, posteriormente, guardada num banco de dados e irá servir como um objeto de comparação nas futuras identificações do indivíduo como é possível visualizar na Figura 1.4 no ponto (4).

Os algoritmos de seleção da face mais neutra e frontal apenas serão aplicados caso a vivacidade do indivíduo for confirmada (representado na **Primeira fase** da Figura 1.6 em (1)).

2.4.1 Expressões faciais

O reconhecimento de expressões faciais tem sido ativamente explorado em computação visual. Com o recente crescimento e popularização da técnica de *Machine Learning (ML)*, o potencial de conceber sistemas inteligentes capazes de reconhecer com precisão as expressões tornou-se uma realidade mais próxima. No entanto, os autores Canedo e Neves [5] afirmam que existem alguns termos que têm um impacto significativo na identificação de expressões de um ser humano, como por exemplo, microexpressões e o contexto envolvente.

Normalmente, o sistema de reconhecimento de expressões faciais é composto por seguintes fases: deteção da face, pré-processamento, extração de características e classificação, como está representado na Figura 2.13. Como apoio a fase de classificação, tem sido utilizado um *Facial Action Coding System (FACS)* desenvolvido por Ekman e Friesen [9] que sintetiza as expressões faciais com base nas *Action Units (AU)*. O *FACS* consiste em 46 *AU*, que descrevem os movimentos faciais. A descrição é baseada na atividade muscular e no efeito que cada *AU* representa nas características faciais.

A técnica proposta por Iftikhar et al. [22] consiste na utilização de um classificador do tipo *Neural Network*, que recebe uma imagem em escala de cinza como *input* e devolve a expressão facial como *output*. Este processo consiste em duas etapas:

1. Pré-processamento da imagem, representado na Figura 2.14 em (1).

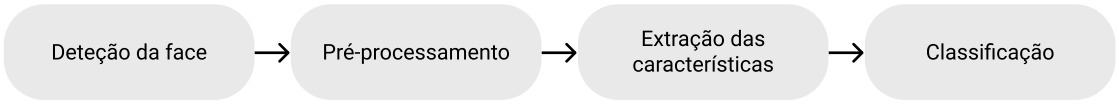


Figura 2.13: Convencional diagrama do sistema de reconhecimento de expressões faciais [5].

2. Treinar ou testar a imagem na *Neural Network* (Figura 2.14 ponto (2)), que por sua vez, classifica a imagem em uma das três categorias: Surpreendido, Normal (Neutra) ou Feliz.

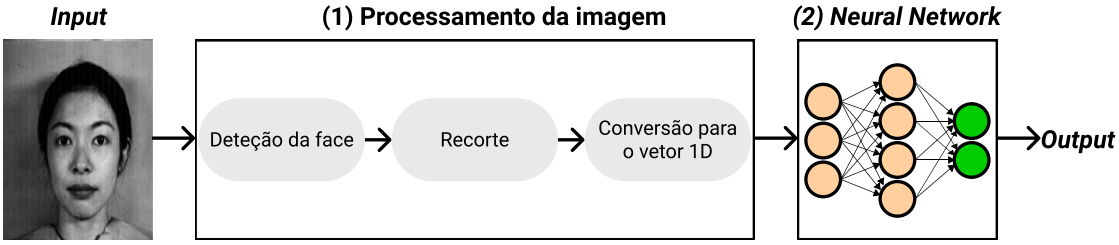


Figura 2.14: Processo proposto por Iftikhar et al. [22]: onde o *input* é uma imagem frontal 2D em escala de cinza, e o *output* é a expressão facial da imagem classificada por *Neural Network*.

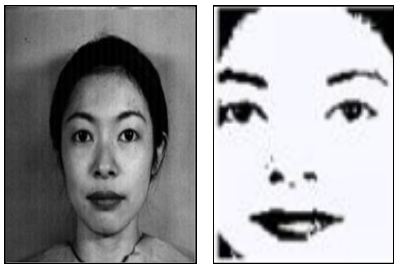
Para efetuar o **Processamento da imagem** é utilizada uma imagem frontal do rosto em escala de cinza de *8-bits*. Com base nessa imagem, efetua-se a deteção da pele que, por sua vez, leva à deteção do rosto. Seguidamente, o rosto detetado é recortado de maneira a obter apenas a área a partir da testa até ao queixo e de orelha a orelha [22]. Este recorte é feito convertendo a imagem para uma imagem binária e efetuando os seguintes passos:

1. Percorrer a imagem na vertical à procura da maior largura (maior número de pixeis brancos contínuos).

2. A procura é interrompida quando a nova largura é metade da largura máxima encontrada anteriormente, pois é sinal que se atingiu a parte das sobrancelhas e não é necessário continuar, porque, a partir desse ponto, a largura será sempre inferior.

3. Segundo o Iftikhar et al. [22], para obter apenas a parte relevante do rosto podemos considerar que a largura é a largura máxima encontrada e que a altura é

1.5 x largura. Aplicando essas medidas a partir da posição inicial da testa obtemos a face recortada como está representado na Figura 2.15 em (b).



(a) (b)

Figura 2.15: Imagem (a) é a imagem recebida como *input*; (b) é a imagem binária que é obtida após o processamento.

Para terminar o processamento da imagem, a imagem recortada é convertida para uma imagem de 32x32 e esta, por sua vez, é convertida para um vetor 1D de dimensão 1024, em que o pixel preto é representado como sendo 1 e branco como sendo 0. Este vetor é utilizado como *input* para a *Neural Network*.

Relativamente a segunda parte dessa técnica, é criada uma ***Multilayer feed-forward Neural Network architecture***. Essa arquitetura consiste em 1024 neurónios na *input layer*, isto é, um vetor unidimensional da imagem binária de dimensão 32x32. Em relação a *output layer*, esta é constituída por 3 neurónios, um para cada categoria de expressão. Os autores afirmam que para treinar a rede, diferentes combinações de *hidden layers* e neurónios por cada *layer* foram testadas. Com base no menor erro, apenas uma *hidden layer* com 80 neurónios foi escolhida para a rede dessa abordagem. Relativamente aos parâmetros, como *learning rate* e *momentum*, também são escolhidos pelo método *hit and trial* e são 0.1 e 0.3 respetivamente. Segundo o Iftikhar et al. [22], para treinar a rede é utilizado o algoritmo *Back-Propagation* e *sigmoid function* é utilizada como função de ativação.

2.5 *Adversarial Machine Learning*

A técnica designada de *Adversarial Machine Learning* é bastante utilizada em *ML* para iludir um modelo utilizando dados de entrada maliciosos. Embora esta técnica possa ser utilizada em diversas situações, ultimamente, tem sido bastante comum a sua utilização na criação de ataques ou provocação de funcionamento indevido nos sistemas de *ML*.

2.5.1 Tipos de *Adversarial Machine Learning Attacks*

Os *Adversarial Machine Learning Attacks* são agrupados em duas categorias, os que provocam erros nas classificações e os que realizam envenenamento de dados [57].

- A introdução de **Erros nas classificações** é o tipo de ataque mais habitual em que os intrusos escondem conteúdo malicioso nos filtros de uma algoritmo de *ML*. O objetivo deste ataque é que o sistema classifique incorrectamente um conjunto de dados específico.

- Por outro lado, o **Envenenamento de dados** ocorre quando um atacante tenta modificar o processo de *ML* ao colocar dados incorretos no conjunto de dados, tornando os resultados menos precisos. O propósito deste tipo de ataque é comprometer o processo de aprendizagem e minimizar a sua utilidade.

2.5.2 Medidas de proteção contra *Adversarial Machine Learning Attacks*

Hoje em dia, não existe uma forma concreta de se defender contra *Adversarial Machine Learning Attacks*; no entanto, a autora Rouse [57] afirma que existem algumas técnicas que podem ajudar a evitar que um ataque deste tipo aconteça. Tais técnicas incluem *Adversarial training* e *Defensive distillation*.

- ***Adversarial training*** é um processo em que exemplos adversariais são introduzidos no treino do modelo e marcados como sendo ameaças. Esta é uma solução de força bruta onde são gerados muitos exemplos falsificados que posteriormente são passados diretamente para o treino do modelo para que este não seja enganado por cada um deles no futuro. Na presente tese optou-se por aplicar esta abordagem para melhorar a robustez dos modelos desenvolvidos.

- ***Defensive distillation*** tem como objetivo tornar o algoritmo de *ML* mais flexível recorrendo a um modelo que prevê os resultados de um outro modelo que já foi treinado anteriormente com propósito de identificar ameaças desconhecidas. Esta abordagem é bastante semelhante a *GAN*, que é constituída por duas redes neurais concorrentes de forma a acelerar o processo de aprendizagem.

2.5.3 Generative Adversarial Networks (GAN)

A *GAN* é uma rede neuronal não supervisionada que consegue ser treinada de forma autónoma através da análise de um *dataset* com o objectivo de criar novas amostras de

dados que, por sua vez, podem ser utilizados para aumentar a robustez dos modelos. A necessidade de desenvolver este tipo de mecanismos surge devido ao facto de que, nos últimos anos, os casos de cibernéticas têm aumentado drasticamente. Por este motivo, as organizações estão a adotar medidas de segurança avançadas para evitar a fuga e a utilização indevida da informação sensível. No entanto, os hackers têm surgido com métodos inovadores para obter e explorar os dados dos utilizadores. Atividades criminosas como roubo da identidade, chantagem ou humilhação das pessoas através das imagens ou vídeos falsos estão a aumentar e são uma grande preocupação. Uma vez que, cada vez mais dados são partilhados na Internet de forma voluntária, sob a forma de imagens ou vídeos, tornou-se fácil recriar este tipo de ataques. Por isso, um dos métodos utilizados pelos intrusos é *Adversarial Attack* que consiste em adicionar informação maliciosa aos dados, como por exemplo, as imagens (Figura 2.16) ou vídeos.

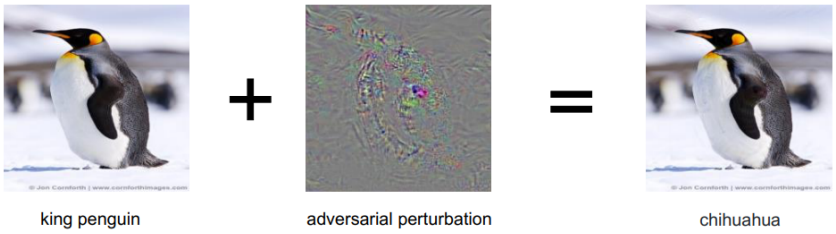


Figura 2.16: Exemplo de um *Adversarial Attack* ao adicionar um simples ruído a imagem. A imagem da esquerda é classificada de forma correcta como sendo um "*king penguin*", a imagem do centro é o ruído que é adicionado a imagem, e a imagem da direita é o exemplo adversarial resultante que está ser classificado de forma incorreta como sendo "chihuahua" [46].

Desta forma, conseguem enganar a rede neuronal, comprometendo o correto funcionamento do algoritmo que, por sua vez, pode resultar na divulgação e comprometimento de informações indesejadas ou mesmo permitir acesso a um determinado sistema. Por essas razões, surgiram as *GANs* que podem ser utilizadas para melhorar o desempenho dos modelos através da criação dos exemplos falsos que, posteriormente, são utilizados durante o treino. Assim, o modelo fica mais robusto a detetar os futuros ataques.

A *GAN* surgiu em 2014 apresentada pelo autor Goodfellow et al. [15], por isso, esta tecnologia pode-se considerar que ainda está na sua fase inicial, mas tem vindo a ganhar muita popularidade devido ao seu poder generativo bem como de discriminação. Um ano depois, os investigadores descobriram que o modelo da *GAN* era instável e exigia um treino bastante complexo. Devido a estas razões, em 2015 o Radford et al. [32] propôs uma versão melhorada da arquitectura da *GAN*, designada de *DCGAN*. Tal como o nome sugere, a melhoria aplicada foi sobre a arquitetura original da *GAN*, passando esta a

ser composta pelas redes convolucionais profundas (*CNNs*). Para além destas alterações, outras melhoria foram implementadas, como por exemplo, remoção das *pooling layers* e das *up-sampling layers* ou utilização do *Batch Normalization algorithm* para resolver o problema de *vanishing gradient*.

2.5.4 Arquitetura e o Funcionamento das *GANs*

A arquitetura da *GAN* é constituida em duas partes, uma com objetivo de gerar novos dados plausíveis (Gerador) e outra com a responsabilidade de classificar estes novos exemplos tendo também em conta os dados reais (Discriminador), como é possível visualizar na Figura 2.17.

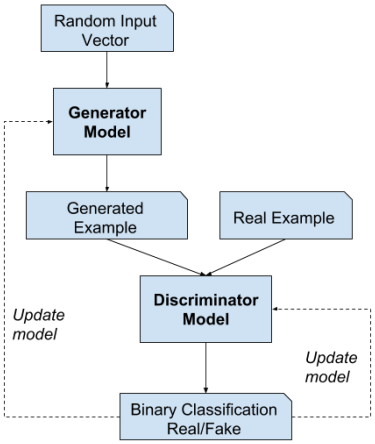


Figura 2.17: Arquitectura da *Generative Adversarial Network*.

Segundo os criadores desta rede e das suas extensões [15, 32], o treino desta arquitetura começa por, em primeiro lugar, treinar o Discriminador com os dados reais que podem ser, por exemplo, imagens, vídeos ou algum tipo de sinais e com os dados produzidos pelo modelo Gerador corrente. Seguidamente, depois de atualizar o modelo Discriminador, é treinado o Gerador tendo em conta os resultados produzidas pelo Discriminador ao classificar os dados reais e os gerados. Deste modo, dependendo das classificações atribuídas aos dados gerados, os pesos do Gerador são atualizados de forma a produzir dados cada vez mais realísticos.

Uma grande vantagem desta arquitetura é que ambos os modelos vão treinando iterativamente um ao outro enquanto procuram a cada iteração melhorar o seu próprio modelo. Segundo o Inkawhich [54] o treino prosegue até ao ponto em que o Discriminador é enganado na maioria das vezes, o que significa que o modelo Gerador está a

conseguir produzir exemplos bastante credíveis. Depois do treino, o modelo generativo pode ser utilizado para criar inúmeras novas amostras falsas que vão permitir melhorar a robustez do modelo numa data tarefa.

2.5.5 *GANs* na Biometria

2.5.5.1 Impressão Digital

Os autores do artigo [30] afirmam que, nos últimos anos, os algoritmos de pesquisa de impressões digitais têm sido bastante limitados devido à falta de dados disponíveis ao público. Essa falta deve-se ao facto de que os regulamentos governamentais sobre a privacidade[7] começaram a proibir a partilha de conjuntos de dados de impressões digitais. Para resolver este problema, o Mistry et al. [30] recorreu à utilização de uma *GAN* para produzir um conjunto de dados constituído por 100 milhões de imagens de impressões digitais. Mais especificamente, utilizou-se a versão melhorada da *Wasserstein GAN (WGAN)* [16] que se deu o nome de *I-WGAN*. Ao aplicar a aboragem proposta por este artigo, conseguiu-se criar impressões digitais muito mais realistas do que as que são criadas recorrendo aos algoritmos tradicionais, o exemplo desta melhoria está representado na Figura 2.18.

Figura 2.18: Comparação entre as impressões digitais criadas utilizando diferentes métodos (a, b, c) e (d) produzida com recurso a *GAN* [30].

2.5.6 Outros exemplos de utilização das *GANs*

2.5.6.1 *Data Augmentation*

Por vezes não é possível criar um conjunto de dados suficientemente significativo para treinar adequadamente o modelo. Desta forma, para diversificar o conjunto de dados são utilizadas técnicas de ampliação que permitem gerar diferentes versões de dados existentes. Normalmente, esta ampliação é feita recorrendo as operações como a inversão, rotação, entre outras. No entanto, as *GANs* têm a capacidade de criar novos exemplos

[7] https://bit.ly/2YD4e4A

a partir dos dados existentes de uma forma mais sofisticada. Como por exemplo, é possível através das *GANs* alterar as imagens existentes de forma a inserir ou remover os óculos de sol ou o sorriso ou, até mesmo, produzir imagens completamente diferentes [21], criando, desta forma, mais e mais diversificados exemplos de imagens como está apresentado na Figura 2.19.

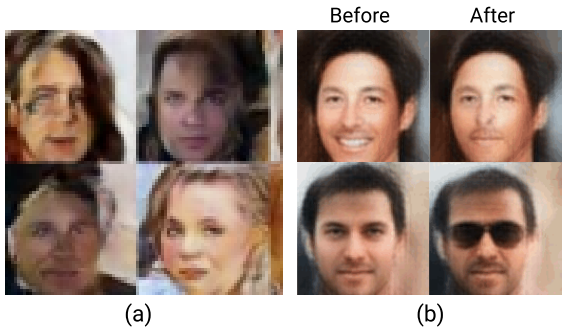


Figura 2.19: Criação de novos exemplos através da *GAN*: (a) geração de imagens de faces com aspecto realista de pessoas que não existem; (b) alteração das imagens existentes no conjunto de dados [21].

De forma análoga, os investigadores da *NVIDIA* conseguirar demonstrar como é que esta técnica de ampliação dos dados pode ser utilizada para aumentar o número de imagens de lesões do fígado para melhorar o desempenho do modelo. Durante esta experiência, utilizaram-se dois métodos diferentes de aumento de dados para avaliar o benefício das *GANs*. No caso do aumento habitual dos dados (rotações e inversões), o desempenho da classificação resultou em 78,60% de *sensitivity (TPR)* e 88,40% de *specificity (FPR)*. Por outro lado, quando as *GANs* foram utilizadas para aumentar o número de dados sintéticos, houve um aumento na *sensitivity* e *specificity*, atingindo 85,70% e 92,40% respectivamente [11].

Concluiu-se, desta forma, que a utilização dos dados originados pelas *GANs* é uma forma bastante eficiente para melhorar a robustez dos modelos.

2.5.6.2 Criação de dados unidimensionais (1D)

As *GANs* podem também ser aplicadas para gerar dados temporais (dados unidimensionais), como por exemplo, sinais cerebrais electroencefalograficos [17], áudio artificial [8] ou electrocardiogramas [7, 13, 48].

Citando o autor Hartmann et al. [17], a geração dos dados temporais é frequentemente abordada utilizando modelos autoregressivos como *WaveGAN* [31] no entanto, as redes neurais convolucionais foram utilizadas na solução apresentada. Por um lado,

porque a maioria dos casos de estudo sobre as *GANs* utilizam as arquiteturas das *DCGANs* [32] que são baseadas em *CNNs*. Por outro lado, porque as estruturas das *CNNs* permitem uma melhor interpretação, o que é muito importante nos sinais cerebrais no contexto neurocientífico ou clínico [17].

Como resultado desta experiência, conseguiu-se gerar através da *DCGAN* os sinais cerebrais electroencefalograficos praticamente indistinguíveis dos reais, como é possível observar na Figura 2.20.

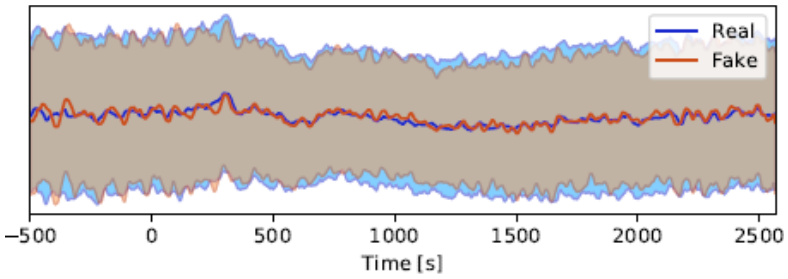


Figura 2.20: Comparação da distribuição dos valores em cada ponto no tempo entre o sinal cerebral electroencefalografico real e o sinal gerado [17].

Por sua vez, ao realizar o teste da *accuracy* observou-se que o modelo conseguiu, de facto, aprender a classificar de forma correta os sinais, tendo atingido uma valor bastante positivo de 91,25%.

Capítulo 3

Redes Convolucionais *1-D* para Deteção da Vivacidade

Os dados de entrada para realizar a distinção entre uma pessoa real e um impostor são sinais de pulsação obtidos através do método chamado *EVM*. A seguinte Figura 3.1 ilustra o processo subjacente à extração do pulso através de *EVM*, um sinal *1-D*, que será utilizado nesta tese.

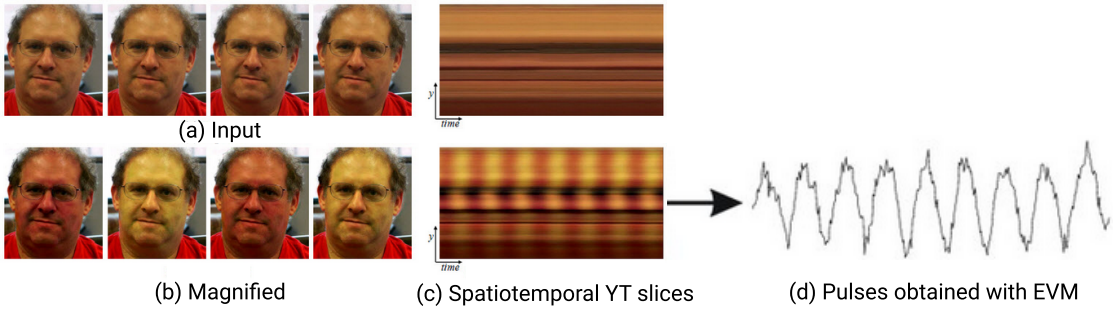
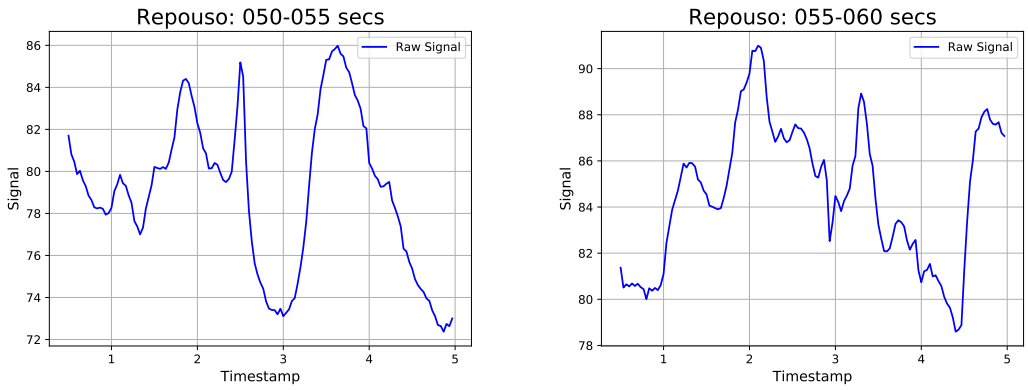


Figura 3.1: Um exemplo da utilização da *EVM framework* para visualizar a pulsação humana. (a) Quatro *frames* do vídeo original. (b) Os mesmos quatro *frames* mas amplificados. (c) Passagem vertical dos dados de entrada (em cima) e dos dados de saída (em baixo) mostra a variação periódica da cor ao longo do tempo [29, 45].

Recapitulando brevemente o problema, é fornecido como dados de entrada um vídeo de 5 segundos, que é uma sequência de *frames* (Figura 3.1a). Seguidamente, a estes *frames* é aplicado o método de *EVM* (Figura 3.1b) para estimar o sinal de pulsação (Figura 3.1d).

E por fim, depois de obtido o sinal de pulsação, é o objetivo desta tese decidir se este provém de um indivíduo real ou se se trata de um ataque de apresentação. Para isso, recorreu-se à utilização de algoritmos de *ML*, mais concretamente redes convolucionais profundas.

Visto que se trata de um sinal, concluiu-se rapidamente que os dados são unidimensionais (*1-D*) em que cada sinal é constituído por 128 valores dispersos ao longo do tempo. Como é possível visualizar na Figura 3.2, o sinal contém características naturais, como por exemplo, variação do período, amplitude variável, início indefinido e a mais óbvia neste contexto é que altera de sujeito para sujeito.



(a) (b)

Figura 3.2: Exemplo concreto dos sinais de pulsação de um indivíduo real que é necessário classificar.

Devido a estas propriedades intrínsecas, decidiu-se aplicar operações convolucionais para que fosse possível detetar padrões que permitissem efetuar esta diferenciação, o que por usa vez, motivou a utilização de diferentes camadas nas redes neurais desenvolvidas. Dito isto, implementou-se e testou-se três modelos convolucionais diferentes, tais como: *CNN* [14], *CNN* com *Residual block* [18] e *TCN* [3].

3.1 *Convolutional Neural Network*

A *CNN* [14] é uma rede neural profunda computacionalmente eficiente que tem demonstrado um excelente desempenho em diversas tarefas que envolvem sinais *1-D*, que é o caso do problema proposto para esta tese. No entanto, pode também ser utilizada em aplicações que lidam com dados compostos por imagens, como por exemplo o conjunto de dados *Image Net*, *computer vision* ou *Natural Language Processing (NLP)*. A principal

vantagem em utilizar uma *CNN* em comparação com as outras redes neurais é que esta permite aprender automaticamente os filtros mais adequados para classificar os sinais.

Observando o esquema do problema proposto para esta tese na Figura 3.3, é possível visualizar que a *CNN* recebe como dados de entrada uma sequência de valores, que representam o sinal de pulsação estimado e, de modo independente, deteta as caracterísitcas distintivas que permitem efetuar a classificação binária.

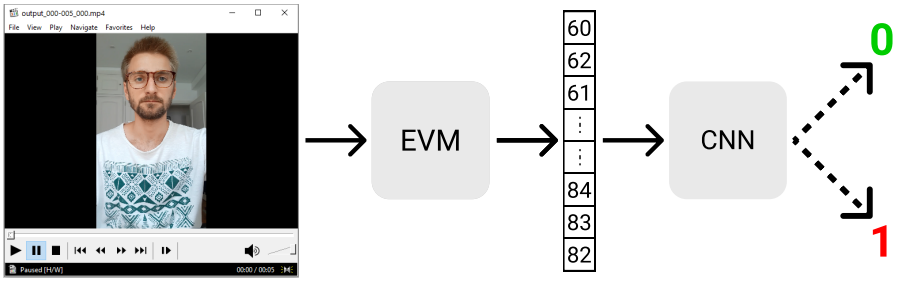


Figura 3.3: Todo o processo necessário para efetuar a classificação binária do sinal de pulsação estimado.

Por sua vez, esta classificação corresponde à deteção de um ataque de apresentação (ausência da vivacidade → ataque de apresentação → 1) ou presença de um indivíduo real em que existem indícios de vivacidade → 0. Para que a classificação do sinal seja correta, é necessário identificar um conjunto de características, como já foi referido anteriormente, e, para isso, utilizou-se um conjunto de operações de convolução e de *pooling* que acabam por constituir a maior parte desta rede neural. Todos os modelos *CNN*, cujo o objetivo é a classificação de um sinal ou algo semelhante, seguem uma arquitetura idêntica à que é apresentada na Figura 3.4.

Para efetuar a classificação do sinal de pulsação, recorreu-se a utilização de duas *Convolutional layers*, duas *Max Pooling layers*, uma *Flatten layer* e uma *Fully Connected layer*.

Relativamente a *Convolutional (Conv) layer*, é a camada mais importante e é a que consome mais tempo de computação dentro de uma rede devido à elevada quantidade de cálculos que é necessário realizar. Habitualmente, esta é a primeira camada numa *CNN* onde são aplicadas designadas operações convolucionais, que consistem em operações lineares sistemáticas entre os dados de entrada e os filtros que vão sendo aprendidos. Como resultado destas operações, são produzidas as chamadas *feature maps*. Desta forma, dado o sinal de entrada utilizado na corrente tese, aplicou-se uma *Conv layer* com intuito de detetar padrões específicos que permitissem distinguir o sinal de pulsação proveniente de um indivíduo genuíno de um ataque de apresentação. Esta deteção

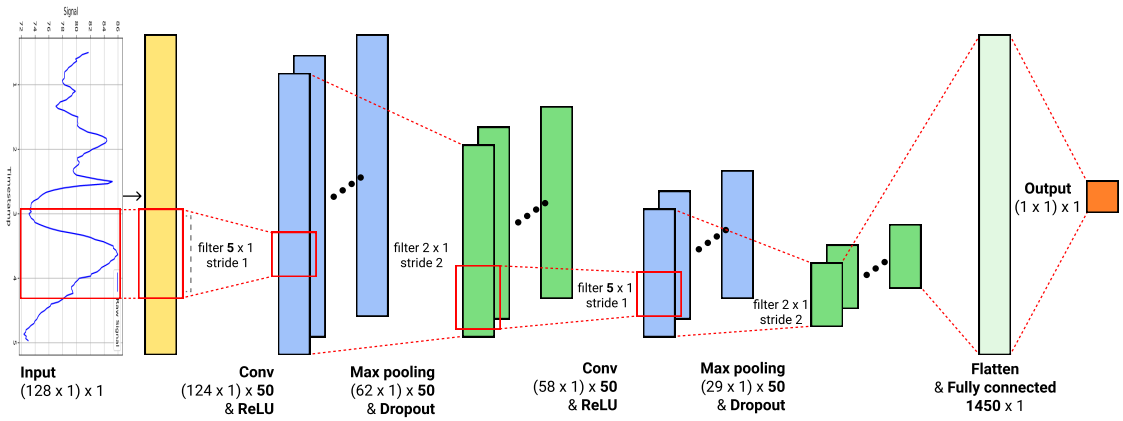


Figura 3.4: Arquitetura do modelo *CNN* utilizado para classificar o sinal de pulsação estimado.

apenas é possível devido a uma variedade de filtros que têm a capacidade de identificar características diferenciadoras ao longo do sinal de entrada. A grande diferença entre as duas *Conv layers* representadas na Figura 3.4 é que a primeira permite apenas detetar padrões de baixo nível (*low-level features*), enquanto que a segunda, estando numa fase mais avançada da rede, conseguirá potencialmente captar características de mais alto nivel (*high-level features*).

Por outro lado, visto que as *feature maps* produzidas pelas *Conv layers* são sensíveis à localização das características nos dados de entrada, decidiu-se realizar uma redução das *feature maps*. Para isso, aplicou-se sempre uma *Max Pooling layer* após uma *Conv layer* que tem o efeito de tornar as *feature maps* reduzidas mais robustas a mudanças de localização das características no sinal de pulsação. Pois, assim as *feature maps* apenas contêm as características presentes com mais frequência num dado sinal de entrada. Esta camada é essencial no problema da presente tese, pois o mesmo padrão poderá não ocorrer sempre no mesmo instante e, desta forma, consegue-se garantir que, mesmo estando numa localização ligeiramente diferente, é possível detetá-lo.

A última etapa de uma *CNN* é um classificador designado também por *Dense layer*, que neste caso, é composta por uma *Flatten layer* e uma *Fully Connected layer*. Sendo um *Artificial Neural Network (ANN) classifier*, é necessário ter as características individuais que, por sua vez, constituem o *feature vector*. Para isso, converteu-se as características extraídas do sinal através da parte convolucional de uma *CNN* num *1-D feature vector* utilizando a *Flatten layer*. Por fim, utilizou-se uma *Fully Connected layer* seguida de uma função de ativação chamada *sigmoid* com intuito de efetuar uma classificação binária dos dados presentes no *feature vector* como sendo um sinal genuíno (0) ou um ataque de

apresentação (1).

Para que o melhor modelo fosse selecionado e aplicado no teste, utilizou-se durante o treino e validação uma função de custo designada por *Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss)* que permite calcular o *loss error*. Tendo este erro sido calculado, considerou-se como sendo o melhor modelo aquele que obteve o menor *loss error* durante a validação. Este processo de determinação do erro e da seleção do melhor modelo aplicou-se, do mesmo modo, aos restantes dois modelos apresentados nas próximas Secções 3.2 e 3.3.

3.2 *CNN* com *Residual block*

Tal como o título sugere, este modelo é uma combinação de uma *CNN*, apresentada na secção anterior, com um *Residual block*, como está representado na Figura 3.5.

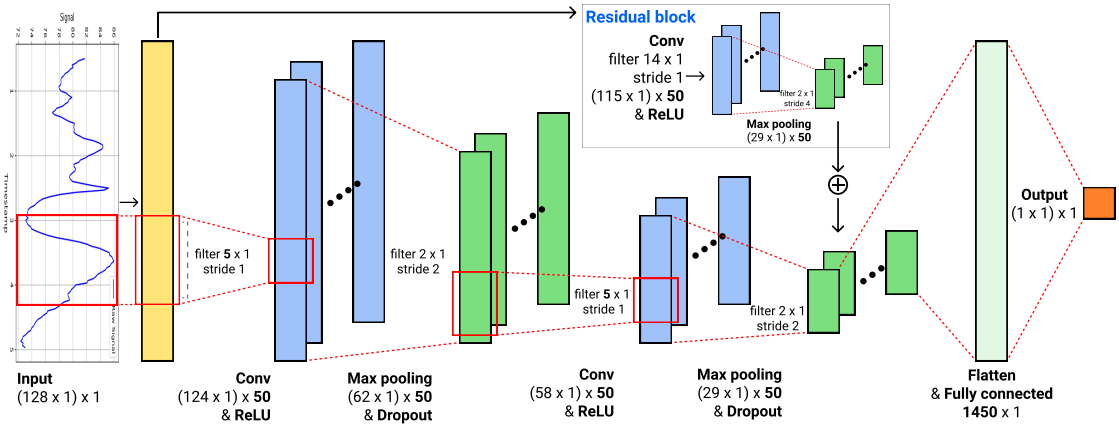


Figura 3.5: Arquitetura do modelo *CNN* com um *Residual block* utilizado para classificar o sinal de pulsação estimado.

Recentemente, vários estudos têm apresentado a utilização dos *Residual blocks* [3, 14, 43] em *Deep learning* uma vez que se tem conseguido atingir um melhor desempenho em determinadas tarefas. Deste modo, decidiu-se implementar esta vertente da *CNN* para verificar se existe de facto uma melhoria ao executar a tarefa da presente tese em comparação com a *CNN* simples apresentada anteriormente na Secção 3.1.

A introdução destes blocos tem uma grande vantagem para uma rede neural profunda, pois permite que os modelos sejam muito mais profundos sem o risco de ocorrer o problema de degradação dos gradientes, uma vez que, as camadas aprendem as modificações ao elemento x em vez de toda a transformação $\mathcal{F}(x)$ como está representado formalmente na Equação 3.1.

$$o = Activation(x + \mathcal{F}(x)) \tag{3.1}$$

Para introduzir o *Residual block* apresentado na Figura 3.5, teve-se o cuidado de escolher os parâmetros nomeadamente, dimensão dos filtros e os *strides*, para que fosse possível obter os dados cujo dimensão é igual à que é produzida pela rede até ao momento da soma da Equação 3.1. Apenas desta forma foi possível realizar a adição e garantir o correto funcionamento e utilização de um *Residual block*. Tendo esta alteração, as redes profundas nomeadamente as *CNNs*, têm demonstrado uma melhoria no desempenho em algumas tarefas. Dito isto, conseguiu-se de facto verificar uma pequena melhoria no desempenho ao utilizar uma *CNN* com *Residual block* em vez de uma *CNN* simples.

3.3 *Temporal Convolutional Network*

Uma vez que os dados de entrada do problema descrito nesta tese variam ao longo do tempo, considerou-se formular o problema como uma tarefa de **Modelação Sequencial**.

3.3.1 Modelação Sequencial

A tarefa de modelação sequencial é definida pelo autor Bai et al. [3] através de um exemplo generalizado em que dada uma sequência de entrada $x_0,\ldots,x_T$, o objetivo é prever os valores de saída correspondentes $y_0,\ldots,y_T$. No entanto, para realizar esta previsão é imposta uma restrição fundamental, que afirma que, para prever um valor de saída y_t para um dado instante t, é apenas permitido utilizar os valores de entrada observados anteriormente $x_0,\ldots,x_t$ e nenhum valor de entrada do futuro $x_{t+1},\ldots,x_T$. Assim, segundo o artigo [3] a modelação sequencial é definida formalmente através da seguinte igualdade:

$$\hat{y}_0,\ldots,\hat{y}_T = f(x_0,\ldots,x_T) \tag{3.2}$$

O principal objetivo, quando se trata de uma aprendizagem automática no contexto da modelação sequencial, é encontrar uma rede f que minimize a perda entre os valores de saída atuais e os previstos. Visto que, o tipo de dados desta tese enquadra-se na descrição efetuada acima, decidiu-se adoptar a arquitectura do modelo ***TCN***, que foi proposta com o intuito de obter uma rede convolucional profunda para dados sequenciais.

3.3.2 Arquitetura e utilização da *Temporal Convolutional Network*

A *TCN* combina simplicidade, previsão autorregressiva[1] e **memória muito longa**, isto é, a capacidade da rede de olhar muito longe para o passado para realizar as previsões. Este modelo baseia-se em dois princípios fundamentais:

1. As convoluções na arquitetura são causais, o que significa que não existe fuga de informação do futuro para o passado.
2. A arquitetura pode receber como entrada uma sequência de qualquer comprimento e mapeá-la para uma sequência de saída com o mesmo comprimento.

Sabendo isto, analisou-se e concluiu-se que o sinal de pulsação estimado adequa-se perfeitamente aos dois princípios referidos. Para cumprir o primeiro princípio, a *TCN* utiliza convoluções causais, isto é, convoluções em que um valor de saída no instante t é gerado pelos elementos do instante t e anteriores da camada anterior. Por outro lado, para alcançar o segundo princípio, a *TCN* recorre ao uso de uma arquitetura *1-D Fully-Convolutional Network (FCN)*, em que cada *hidden layer* tem o mesmo comprimento que a camada de entrada.

Uma vez que as convoluções causais simples são apenas capazes de olhar para trás para um histórico cuja dimensão é linear com a profundidade da rede, torna-se difícil a aplicação das convoluções causais nas tarefas sequenciais, sobretudo naquelas que requerem um histórico muito longo. No problema desta tese, é necessário considerar na integra todo o sinal de pulsação na classificação, pois os padrões de vivacidade poderão ocorrer em qualquer instante do sinal. Por isso, o Bai et al. [3] propôs uma solução que é implementar as **Convoluções dilatadas**, que permitem aumentar de forma exponencial o *receptive field* que, por sua vez, permite abranger uma maior quantidade de contexto para a tarefa. Este tipo de convoluções é equivalente à introdução de um espaço fixo entre cada dois elementos adjacentes do filtro [3], como é possível observar na Figura 3.6.

Visto que, os sinais de pulsação têm um conjunto de características próprias, que já foram referidas anteriormente, e que variam de indivíduo para indivíduo, isso implica que os padrões de vivacidade possam ser de diferentes tamanhos e ocorrer em qualquer momento, como é possível visualizar na Figura 3.2 apresentada no início deste capítulo. Neste caso, as convoluções dilatadas são essenciais, pois permitem que a arquitetura *TCN* tenha um maior acesso ao histórico (consegue analisar partes maiores do sinal), o que por sua vez, leva a que a probabilidade de capturar esses padrões distintos seja maior.

[1] Quando se tenta prever algum sinal dado o seu passado [3].

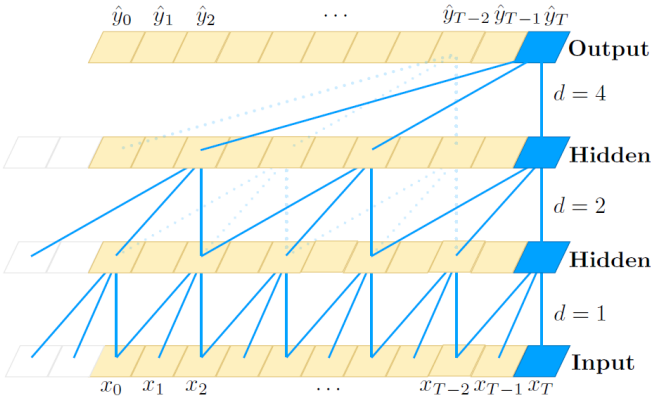


Figura 3.6: Uma convolução causal dilatada com fatores de dilatação d=1, 2, 4 e dimensão do filtro k=3 [3].

Definindo a convolução dilatada de uma forma formal, para uma sequência de entrada unidimensional $x \in \mathbb{R}^n$ e o filtro $f : \{0,\ldots,k-1\} \rightarrow \mathbb{R}$, a operação de convolução dilatada F para o elemento s da dada sequência é definido como:

$$F(s) = \sum_{i=0}^{k-1} f(i).x_{s-d.i} \tag{3.3}$$

Em que d é o fator de dilatação, k é a dimensão do filtro e $s-d.i$ é a direção do passado, isto é, os elementos antecedentes do s com respeito ao fator de dilatação. Como declarado pelo autor Bai et al. [3], utilizando uma dilatação ou um filtro maior permite que os valores de saída no nível superior representem uma gama mais ampla de valores de entrada expandindo, desta forma, o *receptive field* de uma rede convolucional. Isto deve-se ao facto de o histórico de cada camada ser dependente destas duas variáveis e é representado por $(k-1)d$. Quando as convoluções dilatadas são utilizadas, é comum que o valor do parâmetro d aumente exponencialmente com a profundidade da rede, isto significa que, $d = 2^i$ no nível i da rede, como está ilustrado na Figura 3.6. Desta forma, é possível assegurar que existe algum filtro que consegue cobrir cada valor de entrada dentro do histórico, permitindo assim um histórico extremamente grande utilizando redes profundas.

Dada esta explicação, e sabendo que tendo uma sequência de 128 valores ao longo do tempo que se considerou ser uma tarefa de modelação sequencial, permitiu concluir que, a previsão do último elemento da sequência $\hat{y}_{127}$ é calculada tendo em conta todos os elementos do sinal de entrada. Sendo, por isso, considerada a previsão mais fiável de todas, pois garante que filtros foram aplicados considerando todos os elementos do

sinal de entrada. Para que isso fosse possível, teve-se a preocupação em determinar a dimensão da profundidade (número de camadas) e do filtro, para que esta última previsão dependesse mesmo de todos os elementos anteriores, como é possível ver na figura do Apêndice A.

Dito isto, para determinar o número de camadas necessárias utilizou-se a seguinte Equação 3.4, que permitiu concluir que são necessárias exatamente 7 camadas, como está demonstrado na Figura 3.7.

$$N^{o}layers = \log_2(SequenceLength) \tag{3.4}$$

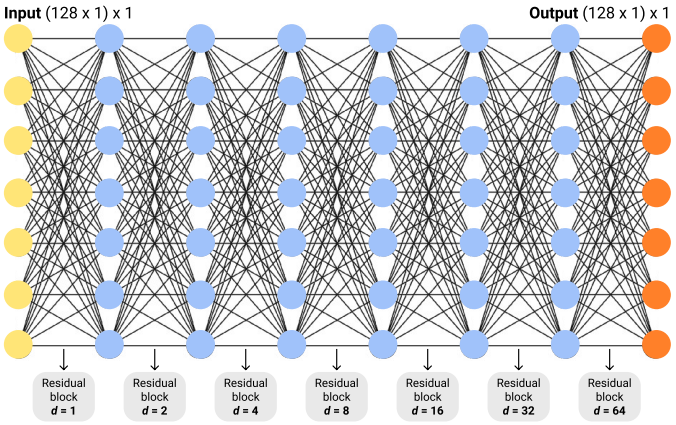


Figura 3.7: Arquitetura do modelo *TCN* utilizado para classificar o sinal de pulsação estimado.

Em relação à dimensão do filtro (*kernel*), recorreu-se a uma abordagem conhecida como tentativa e erro e chegou-se à conclusão que o filtro que permite obter um melhor desempenho é $k = 5$.

Para concluir a arquitetura da *TCN*, cada camada é constituída por múltiplas *hidden layers* (50), ou seja, múltiplos filtros que permitem detetar uma maior variedade de características. Em que no lugar das camadas convolucionais são utilizados ***Residual blocks*** (Figura 3.7 e 3.8) que contêm uma série de transformações $\mathcal{F}$, cujos os valores de saída são adicionados aos valores de entrada do bloco.

Esta abordagem acerca de um *Residual block* é semelhante à que foi descrita na Secção 3.2. Desta forma, toda a *TCN* é constituída por um encadeamento de 7 *Residual blocks*, como é possível visualizar na Figura 3.7.

Apesar das características diferenciadoras, existe uma desvantagem deste modelo,

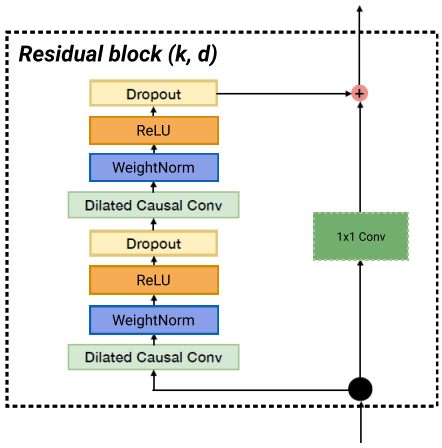


Figura 3.8: *Residual block* de uma *TCN* [3].

é que para obter uma memória muito longa (histórico) é necessário uma rede extremamente profunda ou filtros muito grandes.

3.3.2.1 *CNN* com *Residual block* e a *TCN*

Esta secção surge com intuito de demonstrar e reforçar a ideia sobre a relação que existe entre os modelos *CNN* com *Residual block* e a *TCN* implementados na corrente tese. Como é possível observar na Figura 3.9 ambos os esquemas (a) e (b) representam uma estrutura bastante semelhante.

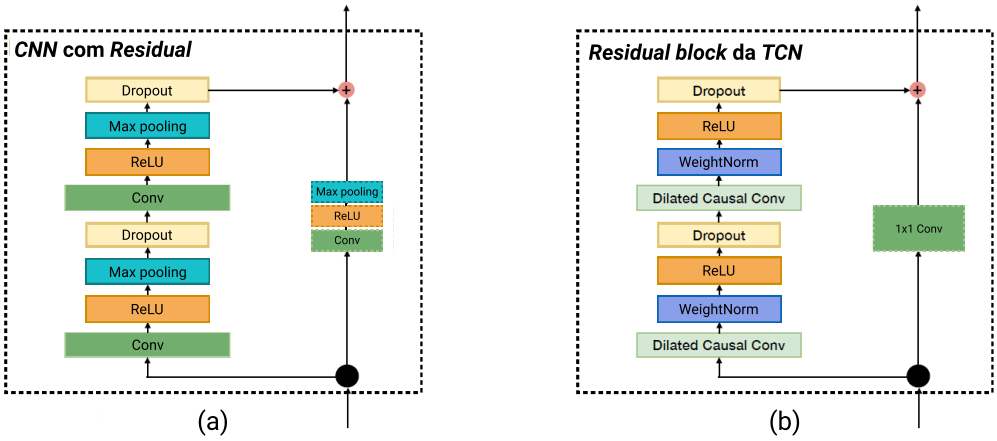


Figura 3.9: Comparação entre a *CNN* com *Residual block* e um *Residual block* de **uma camada** da *TCN*.

No entanto, a estrutura apresentada em (a) representa **toda a *CNN* (parte esquerda**

da figura) incluindo também o *Residual block* (parte direita da figura) enquanto que a representação em (b) demonstra apenas a constituição de **uma camada da *TCN* que é somente um *Residual block***. Devido a estas parecenças, conseguiu-se concluir que a *CNN* com *Residual block* é apenas uma aproximação a uma camada da *TCN*, ou seja, *TCN* com apenas 1 nível. É de notar também que uma das diferenças entre as duas estruturas apresentadas na imagem acima é o tipo das convoluções utilizadas, em que numa *CNN* com *Residual block* utilizou-se as convoluções simples e numa *TCN* convoluções dilatadas pelas razões apresentadas na secção anterior. Para além desta diferença tem-se o facto de que o sinal vai sendo reduzido ao longo da *CNN* + *Residual block* ao contrário do que acontece numa *TCN* que mantém sempre a mesma dimensão do sinal recebido como dados de entrada. A elaboração desta aproximação deve-se ao facto de que é esperado que o modelo *TCN* consiga produzir resultados bastante positivos quando aplicado ao problema descrito na presente tese. Por isso, esta vertente da *CNN* serve para simular uma pequena parte da *TCN* e verificar se é possível atingir um desempenho semelhante.

3.4 Sumário

- De uma forma sucinta, neste capítulo apresentou-se três modelos convolucionais *CNN*, *TCN* e *CNN* com *Residual block* implementados com objetivo de determinar qual deles representa o melhor desempenho na tarefa de deteção da vivacidade.
- Adicionalmente, discutiu-se a relação entre o modelo *CNN* com *Residual block* e o *Residual block* de uma *TCN* descrevendo também, a verdadeira razão pela qual se decidiu criar esta vertente da *CNN*.
- Para que a escolha do melhor modelo fosse realizada, elaborou-se no Capítulo 5 na Secção 5.4 a comparação dos três modelos com variação dos respetivos *hyper-parameters*. À partir da análise dos resultados descritos na secção referida anteriormente, escolheu-se o modelo com melhor capacidade de distinção entre o ataque de apresentação e a apresentação real, uma vez que, é necessário garantir que este seja o mais seguro para o contexto do problema.

Capítulo 4

Geração de Sinais Cardíacos Artificiais

Este capítulo apresenta uma abordagem que permite, de forma automática, criar sinais cardíacos artificiais que, por sua vez, permitem aumentar a robustez dos detectores de vivacidade. Sinais estes que, no contexto do problema, são considerados como sendo sinais extraídos dos vídeos de ataque de apresentação. Desta forma, tendo este mecanismo, designado de **Treino Adversarial**, conseguiu-se desenvolver um modelo generativo que, por sua vez, pode ser aplicado com dois objetivos: (1) verificar se o modelo selecionado como sendo o melhor a distinguir os sinais falsos dos reais, é robusto ao ponto de garantir a segurança dos utilizadores; (2) verificar se o desempenho dos modelo de deteção de ataques melhora ao treinar com as novas amostrar criadas de forma artificial.

4.1 Treino Adversarial

Em vez de optar por apenas adicionar algum ruído aleatório ao sinal para criar um sinal falsificado, decidiu-se utilizar um método mais sofisticado chamado de *DCGAN* [32] que é capaz de gerar sinais cardíacos artificiais utilizando a estrutura representada na Figura 4.1. Esta rede é uma extensão da *GAN* [15], excepto que utiliza camadas convolucionais e convolucionais transpostas no **Discriminador** e no **Gerador**, respetivamente. A *DCGAN*, introduzida há apenas alguns anos, permite a criação de um modelo generativo bastante eficaz que tem sido um tema de muito interesse na comunidade de *ML*.

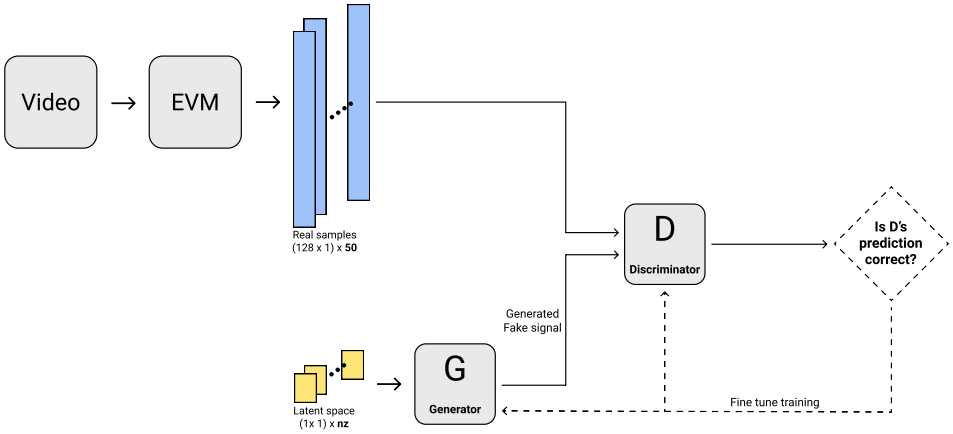


Figura 4.1: Arquitetura do modelo *DCGAN* utilizado para obter o modelo generativo (G) de dados.

Através da rede apresentada na Figura 4.1, conseguiu-se determinar a distribuição dos dados de treino e produzir dados novos (sinais de pulsação falsificados) com uma distribuição bastante semelhante, como é possível visualizar na Figura 4.2.

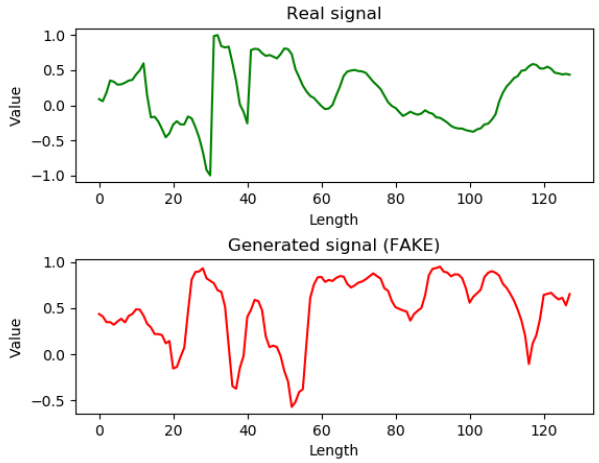


Figura 4.2: Comparação entre o sinal real (em cima) e o sinal falsificado produzido pelo modelo Gerador da *DCGAN* (em baixo).

Desta forma, a geração dos dados não passa por apenas modificar o sinal acrescentando algum ruído, mas sim a criação de um sinal cardíaco artificial praticamente indistinguível do real. Para obter este resultado, como já foi referido, a *DCGAN* utiliza dois modelos distintos, o **Gerador** e o **Discriminador**.

A função do Gerador consiste em, através de uma *seed* aleatória, gerar sinais falsos

que se pareçam com os sinais reais usados durante o treino. Por outro lado, o Discriminador, que é um classificador binário, recebe como dados de entrada tanto as amostras produzidas pelo Gerador, como também as amostras reais do treino e tem como tarefa distinguir ambos os sinais corretamente. Para que fosse possível obter um Gerador de sinais cardíacos, houve a necessidade de colocar as duas redes concorrentes da *DCGAN* numa competição em que ambas se influenciavam uma à outra à medida que se atualizavam iterativamente. Durante o treino, as amostras artificiais são atualizadas e, sendo por isso, o limite de decisão do Discriminador também é ajustado em conformidade, ficando este a aguardar o próximo conjunto de amostras falsas que o tentarão enganar. Desta forma, o Gerador está constantemente a tentar iludir o Discriminador ao gerar falsificações cada vez mais realistas, enquanto que o Discriminador está a agir para se tornar um melhor classificador e conseguir destinguir corretamente os sinais falsos dos reais. Por isso, este "jogo" chega a um equilíbrio quando o Gerador está a gerar sinais cardíacos artificiais perfeitos que parecem ter vindo diretamente do conjunto de treino, e o Discriminador por sua vez fica com 50% de confiança em relação a saída do Gerador se é real ou falsa.

Apesar de a ideia principal da *DCGAN* ser a criação de não uma, mas duas redes concorrentes, um Gerador e um Discriminador, nesta tese considerou-se apenas o modelo Gerador resultante. Esta decisão deve-se ao facto da necessidade de existir um modelo que pudesse pôr à prova as capacidades do modelo de deteção de vivacidade e que permitisse gerar inúmeros sinais cardíacos artificiais com objetivo de verificar se o desempenho dos modelos melhora.

Nas seguintes subsecções são apresentados os componentes que compõem a *DCGAN* e a forma de como estes são treinados com mais detalhe.

4.1.1 Gerador de sinais cardíacos artificiais

Tal como o nome da subsecção sugere, nesta parte apresentou-se a arquitetura do Gerador de sinais cardíacos artificiais que é possível visualizar na Figura 4.3.

O Gerador é composto por camadas convolucionais transpostas, camadas de *Batch Normalization* e por funções de ativação *ReLU*. No fim, aplicou-se uma função não linear chamada *Tahn* para garantir que todos os valores resultantes pertencessem a um intervalo de $[-1, 1]$, uma vez que, essa é a escala dos dados de treino reais.

Como dados de entrada, esta rede recebe um conjunto de vetores com apenas 1 elemento extraído de uma distribuição normal e tenta produzir um sinal cujo a dimensão e a distribuição é exatamente igual aos sinais reais do conjunto de treino. Na prática, esta

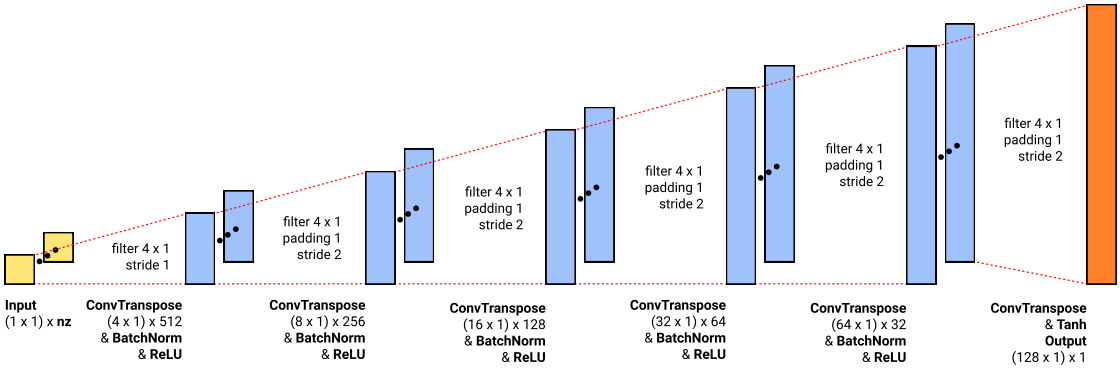


Figura 4.3: Arquitetura do modelo Gerador da *DCGAN* utilizado para produzir sinais cardíacos artificiais.

abordagem apenas é possível devido a utilização do *stride* nas camadas convolucionais transpostas, seguidas de uma camada de *Batch Normalization* e uma função de ativação *ReLU*. É de notar que, os autores Ioffe e Szegedy [23] e Inkawhich [54] reforçam que a existência da camada de *Batch Normalization* a seguir a uma camada de convolução transposta ajuda no fluxo dos gradientes durante o treino e que, em determinadas situações, atua como um regularizador, eliminando a necessidade em utilizar *Dropout*. Permitindo, desta forma, a utilização dos *learning rates* muito mais elevados e ter menos preocupação com a inicialização.

4.1.2 Discriminador não linear para deteção de vivacidade

A arquitetura do Discriminador não linear, apresentada na Figura 4.4, é bastante semelhante em termos dos componentes a arquitetura do Gerador, descrita na subsecção anterior.

O modelo do Discriminador não linear, que é um classificador binário, ao contrário do Gerador, é formado por camadas convolucionais reccorrendo também ao *stride*, seguidas de uma camada de *Batch Normalization* e, por fim, uma função de ativação *LeakyReLU*.

Como dados de entrada, este modelo recebe um sinal de dimensão (128 x 1) e devolve um resultado binário, 1 (sinal artificial) ou 0 (sinal real), recorrendo a função de ativação *sigmoid*. Vale a pena mencionar que, os autores Ioffe e Szegedy [23] e Inkawhich [54] afirmam que é uma boa prática utilizar camadas convolucionais com *stride* em vez de utilizar *pooling* para reduzir a amostra, uma vez que, desta forma a rede consegue aprender a sua própria função de *pooling*.

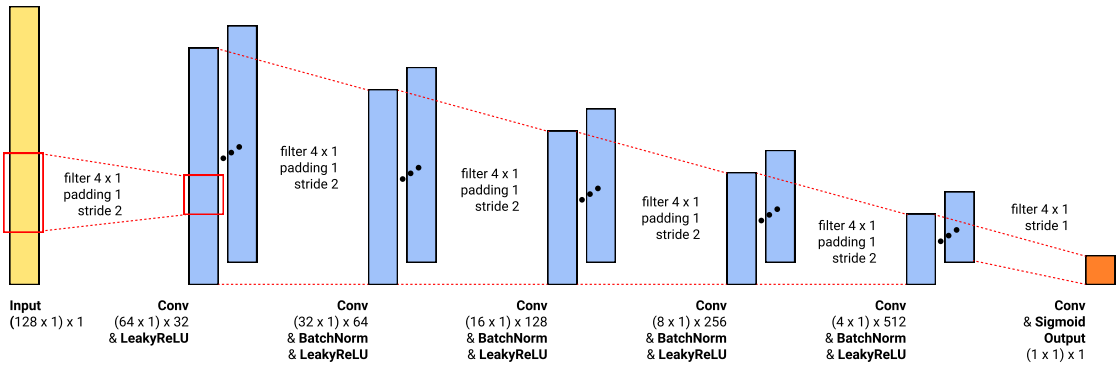


Figura 4.4: Arquitetura do modelo Discriminador não linear da *DCGAN* utilizado para distinguir os sinais cardíacos artificiais dos reais.

4.1.3 Discriminador *TCN* para deteção de vivacidade

Uma vez que o principal objetivo em utilizar a *DCGAN* nesta tese é para obter um bom modelo Gerador de dados, decidiu-se utilizar dois tipos de Discriminadores distintos para que fosse possível realizar uma comparação. Permitindo desta forma concluír qual deles beneficia mais o modelo Gerador final, ou seja, qual deles permite ao Gerador criar dados praticamente indistinguíveis dos dados reais. Os modelos a que se recorreu são a *TCN* que é apresentado no Capítulo 3 na Secção 3.3 e cujo a arquitetura está presente na Figura 3.7 e o modelo não linear descrito na subsecção anterior que, por sua vez, se encontra ilustrado na Figura 4.4.

O resultado e a conclusão a cerca desta experiência encontra-se em Anexo I.

4.1.4 Função de Custo

Antes de introduzir a função de custo que se aplicou durante o treino dos modelos da *DCGAN*, é necessário introduzir algumas notações de cada um dos seus componentes, Discriminador e Gerador.

Notação do Discriminador

- x representa o sinal cardíaco dado como entrada ao Discriminador.

- $D(x)$ representa o modelo do Discriminador que realiza a classificação do sinal x, devolvendo como resultado 1 (sinal artificial) ou 0 (sinal real).

Notação do Gerador

- z representa um vetor extraído de uma distribuição normal.
- *G(z)* representa a rede Geradora que tenta aproximar a distribuição do sinal z a distribuição dos dados de treino reais. Pois, o objectivo do Gerador é exatamente este, estimar a distribuição dos dados de treino (P_{data}) para poder gerar amostras artificiais à partir da distribuição estimada (P_g).
- *D(G(z))* representa o resultado (0 ou 1) da classificação produzida pelo Discriminador ao avaliar um sinal gerado pelo Gerador.

Tendo apresentado as notações, definiu-se a função de custo (Equação 4.1) [15, 54] da *DCGAN*.

$$min_G max_D V(D,G) = \mathbb{E}_{x \sim P_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim P_z(z)}[log(1 - D(G(z)))] \tag{4.1}$$

Analisando esta equação, o Discriminador e o Gerador estão numa *min-max game* em que o Discriminador tenta maximizar o seu resultado ($log(D(x))$), ou seja, realizar de forma correta as classificações, e o Gerador está a tentar minimizar o resultado do Discriminador diminuindo a possibilidade de o sinal gerado ser classificado como tal ($log(1 - D(G(z)))$). Dito isto, concluiu-se que o *loss error* do Gerador diminui quando o Discriminador classifica os sinais falsificados como reais e vice-versa.

No entanto, no início da aprendizagem, quando o Gerador é fraco, o Discriminador consegue rejeitar as amostras com grande confiança, porque são claramente diferentes dos dados de treino reais, ocorrendo desta forma a saturação do ($log(1 - D(G(z)))$). Para resolver este problema, os autores Goodfellow et al. [15] e Inkawhich [54] propuseram uma solução que, em vez de o Gerador tentar minimizar o ($log(1 - D(G(z)))$ é mais benéfico este maximizar o $logD(G(z))$, proporcionando desta forma os gradientes muito mais fortes no início da aprendizagem. Por este motivo, reescreveu-se a equação anterior aplicando esta nova abordagem (Equação 4.2).

$$max_G max_D V(D,G) = \mathbb{E}_{x \sim P_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim P_z(z)}[log(D(G(z)))] \tag{4.2}$$

Esta competição chega a uma solução quando ambos os modelos chegam a um ponto que não conseguem melhorar mais $P_g \approx P_{data}$. Ficando o Discriminador incapaz de diferenciar as duas distribuições, ou seja, $D(x) = \frac{1}{2}$. Contudo, o autor Inkawhich [54] afirma que a teoria da convergência das *DCGANs* está a ser ativamente investigada e, na realidade, os modelos nem sempre treinam até este ponto.

4.1.5 Processo do Treino da *DCGAN*

O processo de treino que se utilizou para treinar a *DCGAN* e obter um bom Gerador de sinais cardíacos artificiais dividiu-se em duas partes principais: na primeira atualizou-se o Discriminador e na segunda atualizou-se o Gerador, como é possível observar no pseudocódigo apresentado em 1.

Algorithm 1 Processo do Treino da *DCGAN* (adaptação do método proposto em [15])

1: **for** number of training iterations **do**

▷ Parte 1

2: Sample batch of m noise samples $\{z^1,\dots,z^m\}$ from noise prior $P_g(z)$

3: Sample batch of m examples $\{x^1,\dots,x^m\}$ from data generating distribution $P_{data}(x)$

4: Update the Discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [logD(x^{(i)}) + log(1 - D(G(z^{(i)})))]$$

▷ Parte 2

5: Sample batch of m noise samples $\{z^1,\dots,z^m\}$ from noise prior $P_g(z)$

6: Update the Generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^{(i)})))$$

7: **end for**

Parte 1 - Treinar o Discriminador

Apenas para recordar que o objetivo do treino do Discriminador é maximizar o seu resultado de realizar, de forma correta, as classificações dos dados falsos e reais. Visto do lado prático, o Goodfellow et al. [15] afirma que o que se pretende é atualizar o Discriminador de forma a aumentar o seu gradiente estocástico e, para isso, é necessário maximizar o $log(D(x)) + log(1 - D(G(z)))$. Dito isto e tendo em consideração a sugestão apresentada pelo Inkawhich [54] em criar separadamente pequenos conjuntos de dados reais e falsos, calculou-se os gradientes em duas etapas.

Primeira etapa

- Em primeiro lugar, criou-se um conjunto de dados reais vindos do conjunto de treino.
- De seguida, o conjunto criado passou-se para o Discriminador para calcular a *loss* ($log(D(x))$) através da função *Binary Cross Entropy Loss (BCELoss)*.
- Para terminar esta etapa, calculou-se os gradientes em *backward pass*.

Segunda etapa

- Por outro lado, na segunda etapa deste processo, no início criou-se um conjunto de dados artificiais produzidos pelo Gerador corrente.

- Tal como na primeira etapa, utilizou-se este conjunto como dados de entrada para o Discriminador para obter a *loss* ($log(1 - D(G(z)))$), recorrendo novamente a função *BCELoss*.

- E, para terminar, acumulou-se os gradientes utilizando de novo o *backward pass*.

Depois destas duas etapas, com os gradientes acumulados dos ambos os conjuntos, invocou-se o *Adam's optimizer* e prosseguiu-se para a parte do treino do Gerador.

Parte 2 - Treinar o Gerador

Tal como afirmado no artigo [15], pretende-se treinar o Gerador de forma a minimizar o $log(1 - D(G(z))$ para conseguir gerar amostras mais realistas. No entanto, como já foi referido acima, Goodfellow et al. [15] provou que esta abordagem não fornece gradientes suficientemente grandes, especialmente no início do processo de aprendizagem, podendo, desta forma, ocorrer o designado *vanishing gradients*. A solução proposta para este problema é, em vez de pretender minimizar o $log(1 - D(G(z))$, maximizar o $log(D(G(z)))$. Em termos de implementação para conseguir esta mudança procedeu-se aos seguintes passos:

- Classificou-se através do Discriminador o conjunto de dados produzido pelo Gerador na **Parte 1**.

- Calculou-se a *loss* do Gerador utilizando as *labels* reais como *Ground Truth* e a função *BCELoss*.

- Computou-se os gradientes do Gerador recorrendo a *backward pass*.

- Por fim, atualizou-se os parâmetros do Gerador utilizando o *Adam's optimizer*.

Esta abordagem em utilizar as *labels* reais como *Ground Truth* no cálculo da *loss* pode parecer um pouco contraditório, mas apenas desta forma, foi possível utilizar a parte de $log(x)$ da função *BCELoss* em vez de $log(1 - x)$, que é o que se pretendia.

O processo de ambas as partes repetiu-se até ao ponto de o Gerador conseguir gerar sinais cardíacos artificiais plausíveis.

4.2 Geração de Sinais Cardíacos Artificiais para Testes de Robustez

Tendo em conta que o corrente conjunto de dados contém um número reduzido de sinais de pulsação falsos, decidiu-se recorrer a *DCGAN* para que novos sinais artificiais fossem gerados de forma automática. O principal objetivo do esquema apresentado na Figura 4.5 é procurar melhorar a robustez do modelo, uma vez que, ao treinar como esses novos dados é esperado que este terá um melhor desempenho ao detetar os futuros ataques de apresentação.

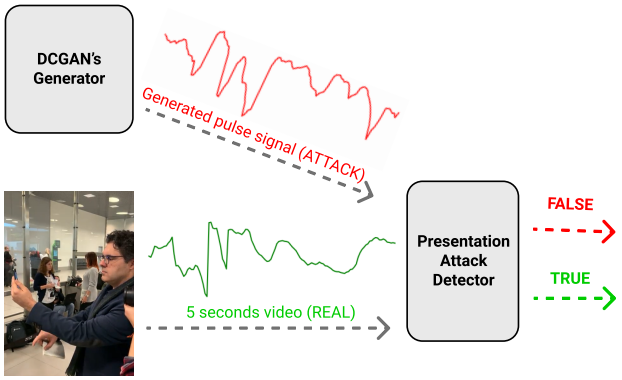


Figura 4.5: Utilização dos sinais de pulsação gerados pelo Gerador resultante da *DCGAN* durante o treino do modelo de deteção de ataques de apresentação.

Por outro lado, o diagrama representado na Figura 4.6, surge com a finalidade de demonstrar como é que o Gerador final da *DCGAN* pode ser aplicado para avaliar a robustez do modelo selecionado para realizar a distinção entre os sinais de pulsação falsificados (produzidos pelo Gerador) e os reais.

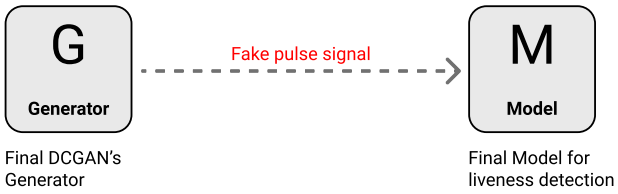


Figura 4.6: Representação superficial da utilização dos sinais de pulsação gerados pelo Gerador resultante da *DCGAN* para testar a robustez do modelo de deteção de ataques de apresentação.

A principal vantagem desta implementação é o facto de ser possível **gerar inúmeros sinais de pulsação falsificados** e verificar qual é o desempenho do modelo escolhido

como sendo o melhor.

Para além da geração de sinais falsos é possível um atacante criar algo mais sofisticado como um vídeo falso a partir da combinação de uma imagem estática e um sinal gerado pela *DCGAN*. Este método encontra-se exemplificado na Figura 4.7 em que é possível visualizar uma fotografia extraída da Internet, um sinal gerado recorrendo a *DCGAN* e várias replicações da imagem original que irão constituir os *frames* de um vídeo. No entanto, ao observar as imagens replicadas com mais atenção é notável que estas têm uma ligeira diferença na região do rosto em relação a imagem original, pois o canal da cor verde sofreu uma pequena alteração ao ser adicionado um valor aleatório do sinal de pulsação gerado a cada pixel. Desta forma, quando o *EVM* receber este vídeo como entrada a estimativa do pulso será muito mais plausível do que se fosse apenas um vídeo a filmar uma imagem estática sem qualquer alteração.

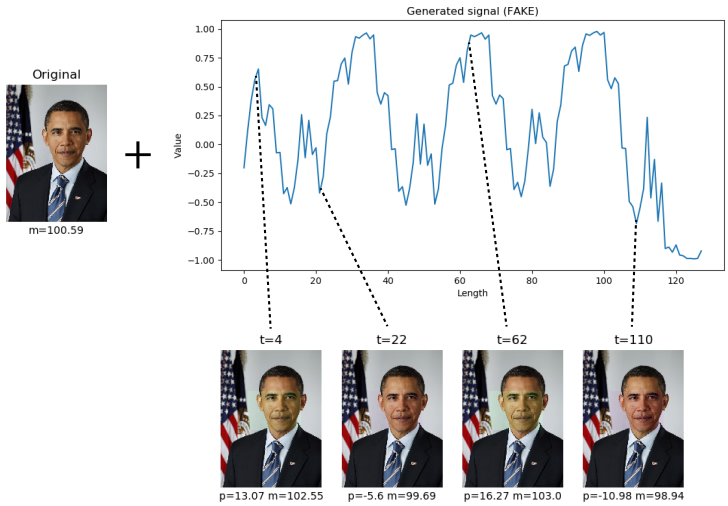


Figura 4.7: Criação do vídeo de ataque de apresentação utilizando sinal de pulso gerado. p - representa o valor do pulso adicionado a cada pixel da região do rosto; m - representa o valor médio da cor verde de toda a fotografia.

4.3 Sumário

- Neste capítulo apresentou-se uma técnica de criação de sinais cardíacos artificiais chamada **Treino Adversarial**, recorrendo especificamente a *DCGAN*.

- Acrescentando, explicou-se todo o processo de treino desta abordagem e quais são os componentes que a compõem.

- Para concluir este capítulo, demonstrou-se como é que o resultado deste processo é utilizado no contexto do problema apresentado nesta tese.

Capítulo 5

Avaliação

Para avaliar os algoritmos propostos, criou-se um conjunto de dados para treinar e testar os algoritmos desenvolvidos, com objetivo de escolher o melhor algoritmo para assegurar a segurança contra ataques de apresentação como a fotografia ou a máscara lisa de papel/plástico. Recolheram-se dados com características e tipos de ataques diferentes, para que os algoritmos fossem testados em diversos ambientes.

O conjunto de dados representa um papel fundamental num processo de desenvolvimento. Este conjunto é imprescindível, pois é através dele que é possível treinar e realizar os testes do sistema implementado e concluir se os resultados obtidos são positivos para o problema em questão. Quanto maior e mais diversificado for o tipo dos dados, mais robustos serão os treinos e os testes e, consequentemente, é possível obter uma noção mais real da performance do modelo, e da sua capacidade de generalização.

5.1 Conjunto de dados

Para que haja uma grande heterogeneidade no conjunto de dados, é necessário combinar dados com características distintas, tais como luminosidade, posição ou cenário em que está inserido, e considerar diferentes tipos de ataque que podem ser, por exemplo, uma simples fotografia da face do indivíduo ou até mesmo uma máscara.

5.1.1 Recolha e Descrição dos dados

Realizou-se uma pesquisa de um conjunto de dados com características acima referidas. Foram feitos vários requerimentos a empresas e instituições que contêm dados com os atributos procurados, mas, infelizmente, nenhum deles obteve sucesso. Portanto, sabendo que se tratam de informações que são consideradas pessoais e que, de forma a não violar regulamentos de protecção de dados, não foi possível obter qualquer conjunto de dados de forma gratuita, ou que tivesse as propriedades necessárias. Assim sendo, decidiu-se recolher um conjunto de vídeos para que fosse possível estudar este problema e, o mais importante, treinar e testar os algoritmos desenvolvidos.

Para recolher maior parte dos dados, utilizou-se um dispositivo móvel *Samsung Galaxy J7 2017* com características presentes na Tabela 5.1.

Tabela 5.1: Características do dispositivo utilizado na recolha de dados.

Desempenho e *Hardware*	**Sistema operativo**	Android 9.0
	Memória Interna	16 GB 3 GB RAM
	Chipset	Exynos 7870 Octa
	CPU	Octa-core 1.6 GHz Cortex-A53
	GPU	Mali-T830 MP1
Ecrã	**Resolução**	1080 x 1920 px
	Densidade de pixeis	~401 *PPI (Pixels per inch)*
Câmara e Vídeo	**Câmara traseira, Principal**	13 MP
	Câmara frontal, *selfie*	13 MP
	Video	1080p@30fps

Todo este conjunto é composto na totalidade por **6 voluntários**. Os dados que se decidiu recolher por cada voluntário são 9 vídeos a filmar a face de um indivíduo real após uma atividade prévia (Tabela 5.1) e 12 vídeos a filmar fotografias em diferentes formatos desse mesmo indivíduo, que é considerado como sendo um ataque (Figura 5.2).

Portanto, na totalidade obteve-se 21 vídeos, de 2 minutos cada, todos eles com características distintas. Ao longo do registo dos vídeos reais, houve necessidade de efetuar uma monitorização contínua do ritmo cardíaco e guardá-lo num ficheiro em formato *.csv*, de forma a posteriormente poder avaliar a performance do módulo de extração de pulso *EVM*, que sai fora do âmbito da presente tese. Esta monitorização foi realizada com ajuda de um *Smart watch Xiaomi Mi Band 5*, representada na Figura 5.1 em (a), e uma aplicação móvel para Android chamada *Mi Band Tools*. Desta forma, obteve-se dois tipos de dados diferentes: os vídeos e os ficheiros com registos de ritmo cardíaco para

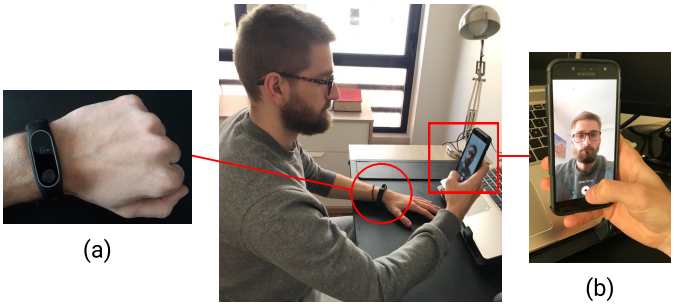


Figura 5.1: Recolha de dados genuínos: (a) *Smart watch Xiaomi Mi Band 5*; (b) *Samsung Galaxy J7 2017*.

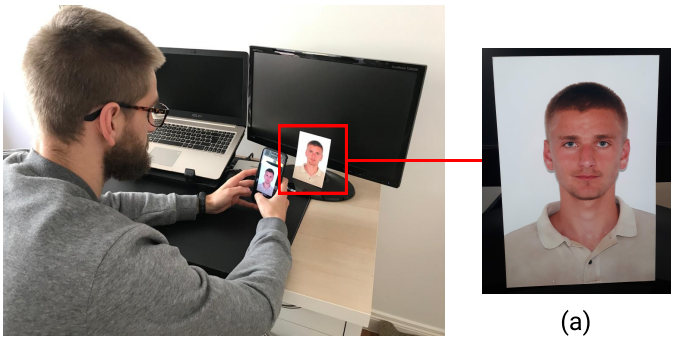


Figura 5.2: Recolha de dados considerados como ataque: (a) fotografia da face do indivíduo.

cada vídeo em que o indivíduo é real. Em relação à monitorização contínua do ritmo cardíaco, cada ficheiro tem informação acerca da hora em que foi feita a medição e o seu respetivo valor, como é possível visualizar na Tabela 5.2.

Tabela 5.2: Monitorização do ritmo cardíaco.

Hora	**Ritmo cardíaco**
13:26:05	62
13:26:01	64
13:25:55	65
13:25:49	61
13:25:43	60
...	...

Todos os vídeos recolhidos apresentam diferentes características, que estão agrupadas na Tabela 5.3, para que o sistema recebesse a maior diversidade de dados possível.

No caso dos vídeos cujo indivíduo é real, para além da iluminação, decidiu-se variar

Tabela 5.3: Plano de criação do conjunto de dados.

	Boa iluminação	**Pouca iluminação**	**Luz apontada para a face/foto**	**Total**
	Atividade prévia			
Vídeos reais	Repouso	Repouso	Repouso	
	Subir escadas	Subir escadas	Subir escadas	
	Andar rápido	Andar rápido	Andar rápido	
Nº vídeos	3	3	3	**9**
	Tipo de foto ataque			
Vídeos de ataques	Expressão facial neutra	Expressão facial neutra	Expressão facial neutra	
	Mexer a foto	Mexer a foto	Mexer a foto	
	Foto do tamanho da face	Foto do tamanho da face	Foto do tamanho da face	
	Foto recortada	Foto recortada	Foto recortada	
Nº vídeos	4	4	4	**12**
				21

também o tipo de atividade física antes da gravação do vídeo, para que o ritmo cardíaco variasse mais e fosse diferente em todos os vídeos. Esta variação é, de uma certa forma, uma simulação de um caso real que pode acontecer no dia a dia de um indivíduo em que este poderá ter estado em repouso, a subir escadas ou a andar rápido antes de efetuar o registo na aplicação.

Por outro lado, em relação aos vídeos de ataque, da mesma forma variou-se a iluminação, mas não foi possível variar a atividade prévia, na medida em que se trata de um artefacto. Assim sendo, variou-se os tipos de foto ataques de apresentação, indicados na Tabela 5.3, tendo sempre em conta os casos que poderão ocorrer no futuro.

Posteriormente, de forma a simular o cenário de utilização real, em que o sistema recebe videos de 5 segundos, sentiu-se a necessidade de cortar os vídeos gravados. Para efetuar esses cortes recorreu-se a dois tipos de janelas deslizantes, 1s e 5s que vão progredindo até perfazer os 2 minutos. Assim, as janelas deslizantes, cujo progressão é apenas 1s do vídeo, dão origem a mais vídeos comparando as janelas de 5s, tendo em conta que existe bastante repetição/sobreposição de conteúdo entre cada vídeo originado. Portanto, tendo sido isso feito por cada um dos 6 voluntários e, dependendo da janela deslizante utilizada, criou-se a seguinte Tabela 5.4 para apresentar a quantidade de dados que é utilizada para treinar, validar e testar os algoritmos implementados. Separação esta que é feita recorrendo a uma função da bibliotéca *sklearn* designada de *train_test_split* que permite misturar os dados, realizar a separação dos mesmos de forma aleatória e tornar a separação reproduzível entre as diferentes execuções.

Assim sendo, a Base de dados contém, na totalidade, 7424 ou 1536 vídeos (respetivamente janela deslizante 1s e 5s). No entanto, reparou-se que o número de vídeos obtidos

Tabela 5.4: Quantidade de dados que se gera por cada voluntário.

	Total vídeos	**Treino (~72%)**	**Validação (~13%)**	**Teste (~15%)**
Janela deslizante 1s	7424	5364	947	1113
Janela deslizante 5s	1536	1110	196	230

com uma janela deslizante de 5s não era suficiente para que os modelos fossem treinados devidamente de forma a produzir resultados plausíveis. Por isso, optou-se por utilizar apenas os vídeos resultantes pela janela deslizante de 1s. Utilizando estes vídeos de curta duração como sendo os dados de entrada do sistema anteriormente desenvolvido no âmbito do projeto *SmartyFlow*, extraíram-se várias propriedades das quais apenas o ***Raw signal*** é que se utilizou como base no desenvolvimento dos algoritmos de deteção da vivacidade. É possível observar algumas das propriedades na Tabela 5.5.

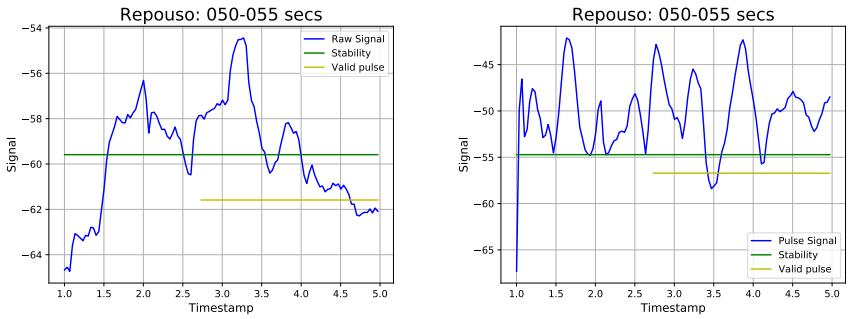
Tabela 5.5: Exemplo de algumas das propriedades extraídas de um vídeo da face do indivíduo.

Timestamp (s)	...	2.84	2.87	2.90	2.94	...
BPM	...	0	0	54.25	54.25	...
Signal stable	...	True	True	True	True	...
Valid pulse	...	False	False	True	True	...
Peaks between 40-240 BPM	...	False	False	True	True	...
Peak amplitudes > 60	...	True	True	True	True	...
Raw signal	...	-65.05	-64.69	-64.42	-63.62	...
Pulse signal	...	-49.03	-47.65	-46.12	-45.49	...

5.1.2 Visualização dos dados

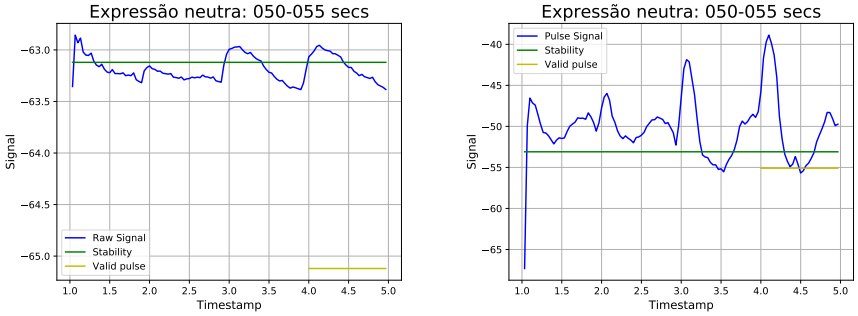
Utilizaram-se algumas das propriedades, presentes na tabela acima, para implementar várias visualizações dos sinais obtidos, como por exemplo, as Figuras 5.3 e 5.4, que permitem obter uma noção sobre o tipo de dados que é necessário classificar. Com base nesses dados, detetou-se os padrões de vivacidade recorrendo aos algoritmos de *Machine learning* apresentados no Capítulo 3.

Observando as duas Figuras 5.3 e 5.4 é notório que o *Pulse signal*, que é o *Raw signal* processado, tanto de uma vídeo genuíno como do vídeo da fotografia são bastante



(a) ***Raw Signal*** (b) ***Pulse Signal***

Figura 5.3: Sinais obtidos a partir do **vídeo da face de um indivíduo genuíno**. Ambos os gráficos representam os respetivos sinais incluindo também o intervalo em que a pulsação é válida e o intervalo de estabilidade do sinal.



(a) ***Raw Signal*** (b) ***Pulse Signal***

Figura 5.4: Sinais obtidos a partir do **vídeo da fotografia da face de um indivíduo**. Ambos os gráficos representam os respetivos sinais incluindo também o intervalo em que a pulsação é válida e o intervalo de estabilidade do sinal.

semelhantes o que poderia tornar a sua distinção muito mais difícil. Por outro lado, analisou-se que o *Raw signal* é bastante diferente, tendo sido este motivo pelo qual se escolheu trabalhar com o mesmo.

5.2 Metodologia de Avaliação

Antes de partir para as metodologias que se aplicaram para interpretar os resultados produzidos pelos classificadores de *Machine learning* desenvolvidos, teve-se o cuidado de definir a chamada *Matriz de confusão* (Figura 5.5) para clarificar e definir o que são cada um dos termos, tais como *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*

ou *False Negative (FN)*.

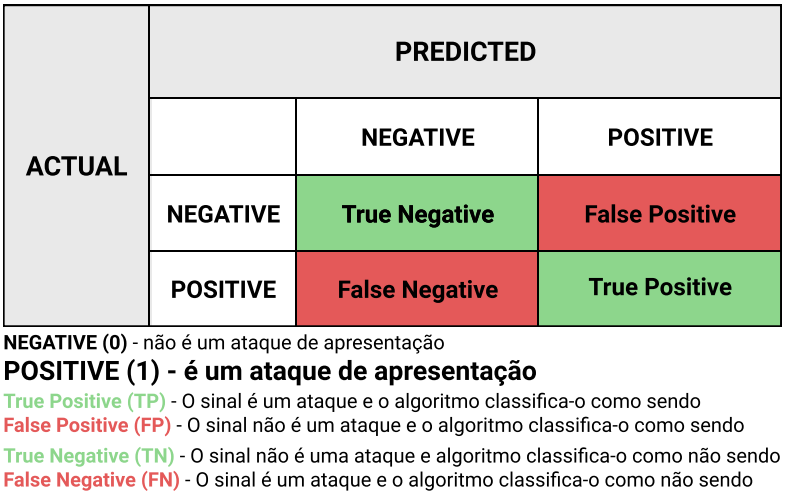


Figura 5.5: *Matriz de confusão* e definição dos seus termos.

Existem várias métricas pelas quais se pode avaliar o desempenho de um classificador, como por exemplo, *Precision* ou *Recall* que são definidas segundo as seguintes Equações 5.1 e 5.2 respetivamente.

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

Nos classificadores desenvolvidos na presente tese para detetar os ataques de apresentação, devido as características do problema **pretende-se que os classificadores terem uma taxa baixa da *Precision* e uma taxa elevada de *Recall* e não vice-versa**. Esta decisão vem do facto de que, no problema em questão, pretende-se garantir a maior segurança possível de cada indivíduo e, por isso, **é menos problemático em termos de segurança os classificadores considerarem um vídeo genuíno como sendo falso do que ao contrário**. Dito isto, tendo a *Precision* baixa não se põe em causa a segurança dos indivíduos, apenas existe uma pequena probabilidade de o vídeo verdadeiro ser considerado como tal e uma grande possibilidade do mesmo ser considerado como um ataque. Por outro lado, mantendo o *Recall* elevado, garante-se que existe uma grande probabilidade de o vídeo real (*Negative class*) ser considerado como tal e que há uma hipótese bastante reduzida de um vídeo de ataque de apresentação (*Positive class*) ser considerado como não sendo.

Deste modo, para avaliar os modelos desenvolvidos recorreu-se às *ROC curves* [10] que têm em conta duas taxas de erro chamadas *TPR* e *FPR* (Figura 5.6) e uma variedade de *thresholds*. Utilizou-se estas curvas com intuito de demonstrar os *trade-offs* entre a *Recall/Sensitivity (TPR)* e a *Specificity (FPR)* para diferentes *thresholds* no conjunto de teste. A *Sensitivity* e a *Specificity* são inversamente proporcionais, assim, quando uma aumenta a outra diminui e vice-versa. Através deste método, determinou-se, também, a *Area Under the Curve (AUC)* que é calculada tendo em conta os valores de *TPR* e *FPR* e que dita o seguinte: **quando maior for esse valor, melhor é o modelo a distinguir os vídeos com indícios de vivacidade e os ataques de apresentação**. Uma observação que é importante mencionar é que existe uma relação entre o *Recall* e a *AUC*, uma vez que, o *Recall* == *TPR* que, por sua vez, influencia o valor da *AUC*.

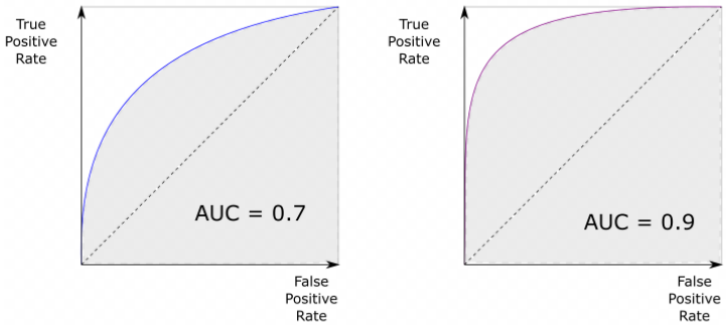


Figura 5.6: Demonstração das *ROC curves* e da variação da *AUC* em relação a *TPR* e *FPR*.

5.3 Implementação

Na implementação dos modelos de deteção de ataques de apresentação recorreu-se a biblioteca *PyTorch*. Relembrando que existe uma grande diferença na implementação dos modelos *CNN* e *TCN*, uma vez que, a *CNN* ao longo da rede vai reduzindo o tamanho do sinal recorrendo a *max pooling* ao contrário da *TCN* que não reduz o sinal a medida que a profundidade da rede aumenta, pois todas as suas camadas são *Fully-Convolutional*. Para que fosse escolhido o modelo com melhor desempenho na tarefa em questão houve necessidade de efetuar o ajustamento dos *hyperparameters* em que se utilizaram as seguintes combinações de valores:

- *dropout* = *{None, 0.05, 0.5}*
- *learning rate* = *{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1, 0.5}*

- *kernel size = {3, 5, 7}*
- *hidden layers = {25, 50}*
- *batch size = {32, 64, 128}*
- levels = {1, 3, 5, 7} (apenas para o modelo *TCN*)

Para que a execução de todas as combinações fosse efetuada em tempo útil utilizou-se um *cluster* com 4 máquinas, cada uma delas com 2 *Graphics Processing Units (GPU) NVIDIA* recentes. Desta forma, cada execução do modelo com uma combinação de *hyperparameters* representa um *job* que posteriormente é consumido e distribuído pelo ***HTC****ondor* (*High Throughput Computing*), que é um sistema de escalonamento para tarefas de computação intensiva.

5.4 Resultados e Discussão

Relembrando que esta dissertação surge com intuito de desenvolver um modelo com capacidades de detetar os ataques de apresentação que poderão surgir no futuro contexto de utilização. Uma vez que, ultimamente, o roubo da identidade tem se tornado bastante recorrente, principalmente na Biometria, é necessário implementar algoritmos cada vez mais robustos para garantir a segurança dos utilizadores. Por este motivo, esta secção contém os resultados das experiências que envolveram diferentes classificadores desenvolvidos, diferentes abordagens de treino e algumas comparações visuais em forma de gráfico entre esses modelos.

5.4.1 Comparação das Redes Convolucionais *1-D*

Tendo a tarefa proposta em mente, desenvolveram-se três detetores de ataques com arquiteturas distintas apresentadas no Capítulo 3: *CNN*, *CNN + Residual block* e *TCN* com objetivo de analisar qual deles é o mais apropriado para o problema em questão. O principal objetivo da implementação destes três modelos é demonstrar que existe uma evolução gradual entre eles à medida que a complexidade da arquitetura vai aumentando, começando com a *CNN* e terminando com a *TCN*. Na avaliação dos modelos optou-se por apresentar como métricas o *Recall*, o *F1-score* e a *AUC* em que o *Recall-1* e o *F1-score-1* são relativamente a capacidade de deteção de ataques de apresentação, ou seja, deteção da classe 1 (*Positive class*). O *F1-score* foi considerado apenas porque reflete o *AUC da ROC*, ou seja, quando *F1-score* tende a aumentar o *AUC* também aumenta.

Assim sendo, após os respetivos treinos em que se utilizou como dados de entrada os sinais obtidos dos vídeos reais e os falsos (filmagens das fotografias) elaborou-se a Tabela 5.6. Tabela esta que contém apenas alguns resultados dos algoritmos desenvolvidos com diferentes combinações de *hyperparameters*, permitindo, desta forma, concluir qual deles é que apresenta um melhor desempenho ao executar a tarefa.

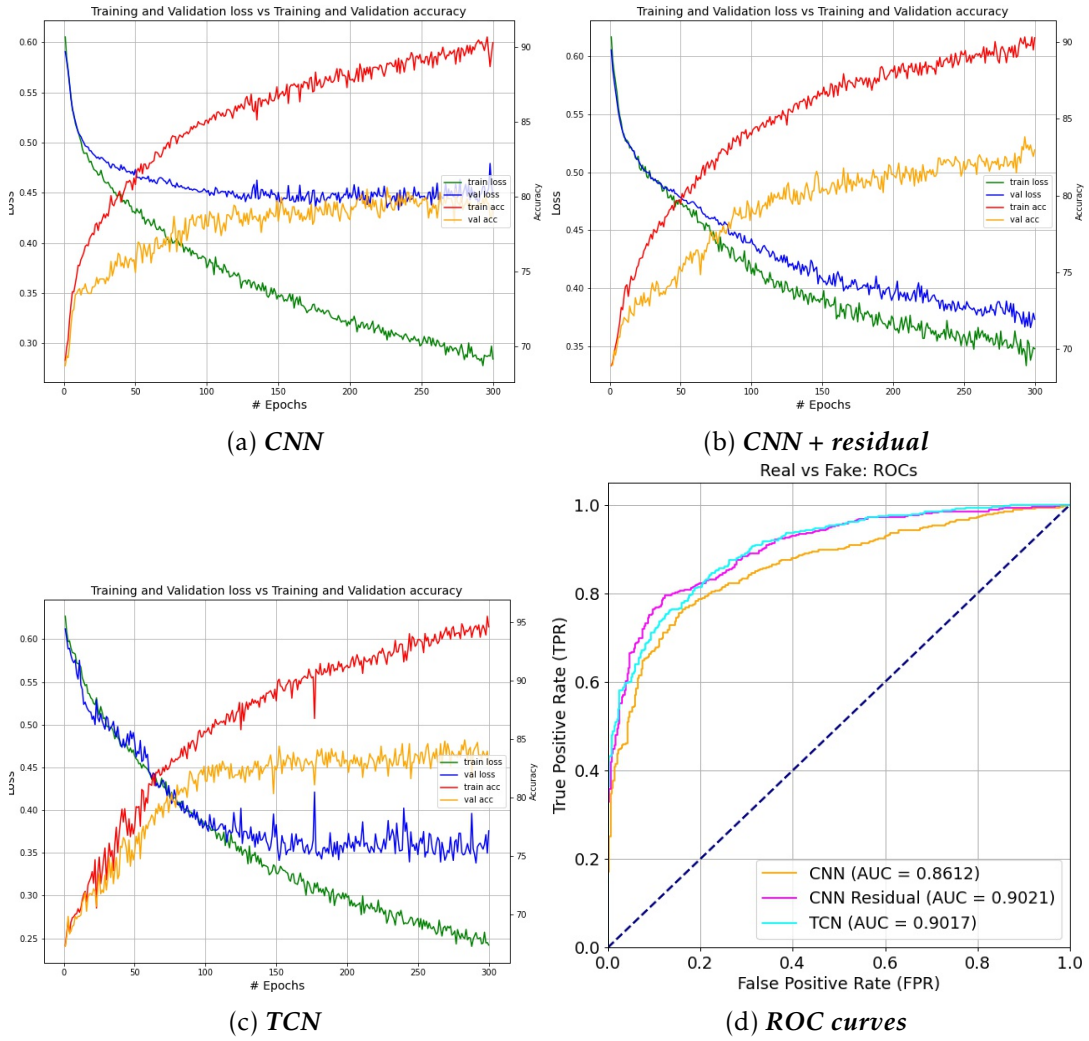
Tabela 5.6: Modelos implementados com diferentes parametrizações e a sua respetiva performance (em percentagem %) na tarefa de deteção de ataques de apresentação.

	Learning rate	**Filter size**	**Hidden units**	**Batch size**	**Dropout**	**Recall-1**	**F1-score-1**	**ROC AUC**
CNN	0,0001	3	25	128	0,5	82,32	79,79	78,10
	0,005	5	25	64	0,5	84,89	81,53	81,05
	0,001	3	25	64	0,5	87,99	83,59	84,10
	0,001	5	50	64	0,5	88,53	84,81	**86,12**
CNN + Residual block	0,0001	7	25	128	0,5	86,23	80,13	77,16
	0,0001	3	50	64	0,5	85,60	81,18	81,44
	0,005	3	25	32	0,5	91,09	86,76	88,22
	0,001	5	50	64	0,5	90,96	87,59	**90,21**
TCN	0,0001	3	25	128	0,5	99,06	80,35	58,10
	0,0005	3	50	64	0,5	88,93	81,66	75,53
	0,001	3	25	32	0,5	83,00	83,67	85,98
	0,001	5	50	64	0,5	89,47	87,70	**90,17**

Analisando os resultados da tabela, reparou-se que existe um conjunto de *hyperparameters* que se destaca em todos os modelos implementados. Conjunto este que é composto por *Learning rate (lr) = 0,001; Filter size (k) = 5; Hidden units = 50; Batch size = 64; Dropout = 0,5* e *levels = 5* no caso da *TCN*. Estes são os valores que permitem de facto, obter resultados significativamente melhores em relação as outras combinações de *hyperparameters* em todos os modelos. Desta forma, concluiu-se que, para detetar os ataques de apresentação com mais eficácia, é mais adequado a utilização de *lr* significamente baixo para que o modelo não seja muito alterado em resposta ao *loss error* estimado, sempre que os pesos do modelo são atualizados. Por outro lado, é recomendavel também que a dimensão do filtro (k) seja relativamente elevada para que cada instante tenha mais informação sobre os seus vizinhos, permitindo desta forma a extração de *low-* e *high-level features* mais significativas.

Dito isto, observando os registos de cada modelo em questão, teve-se, também, a preocupação de encontrar um balanço entre o *Recall-1* e a *AUC*. Uma vez que, é muito importante que o *Recall* seja elevado para garantir que nenhum ataque de apresentação seja considerado como não o sendo, mas, por outro lado, é fundamental garantir que o algoritmo seja bastante robusto a realizar a destinção entre as duas classes para obter

uma boa usabilidade do sistema, sendo, por isso, essencial que a *AUC* também seja elevada. Deste modo, para cada um dos modelos apresentados na tabela, considerou-se como sendo o melhor registo a última combinação de *hyperparameters* da Tabela 5.6.



(a) ***CNN*** (b) ***CNN + residual***

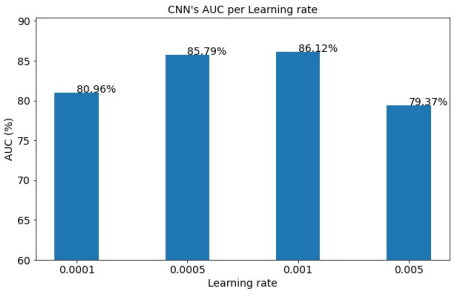
(c) ***TCN*** (d) ***ROC curves***

Figura 5.7: (a, b) Exemplos de uma das execuções de cada um dos modelos com o respetivo decréscimo da *loss* e do aumento da *accuracy*. Em (d) estão representadas as *ROC curves* dos respetivos modelos.

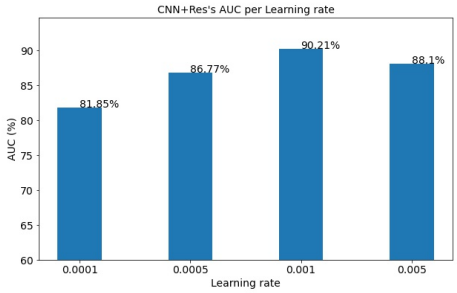
É notório que, ao longo do treino de cada um dos modelos, representados na Figura 5.7, existe uma melhoria constante na sua performance ao detetar os ataques de apresentação. No entanto, observando os resultados apresentados tanto na Tabela 5.6, como também o gráfico das *ROC curves* 5.7d, é notável que a evolução gradual que houve da *CNN* (*AUC* = 86,12) para a *CNN* com *Residual block* (*AUC* = 90,21) melhorou,

de facto, o desempenho e, por isso, se verifica a vantagem da utilização de um *Residual block* apresentada no Capítulo 3. Contudo, existe um outro modelo que se destaca e apresenta resultados equivalentes a *CNN* com *Residual block* que é o classificador *TCN*, uma vez que, consegue manter o *Recall-1* bastante elevado (89,47) e continuar a demonstrar uma excelente performance na distinção de ambas as classes (90,17) (*AUC*). Desta forma, com estes resultados concluíu-se que tanto a *CNN* com *Residual block* como a *TCN* poderiam ser escolhidas neste cenário pois, são modelos que conseguem garantir mais segurança e, daí, são os mais adequado para efetuar a deteção de ataques de apresentação no contexto de aplicação.

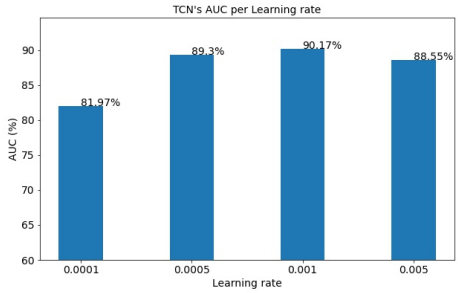
Decidiu-se também analisar a evolução da performance fixando alguns dos *hyperparameters* e variando outros para analisar melhor o comportamento dos modelos. Como resultado desta experiência elaborou-se as seguintes Figuras 5.8 e 5.9.



(a) *CNN's AUC per learning rate*



(b) *CNN + Residual block's AUC per learning rate*



(c) *TCN's AUC per learning rate*

Figura 5.8: *AUC* ao longo da variação do *learning rate hyperparameter* para cada modelo.

Observando os resultados da figura acima, confirma-se que de facto o melhor *learning rate* para todos os modelos é de *0,001*, pois permite que o modelo consiga distinguir melhor ambas as classes.

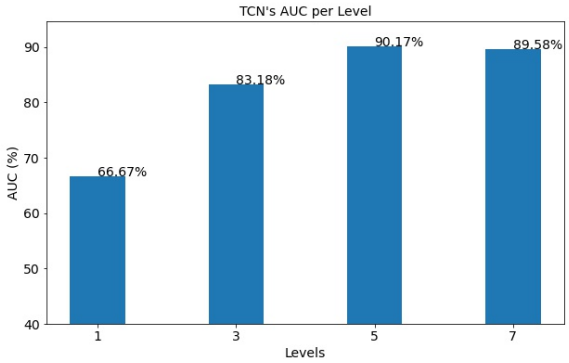


Figura 5.9: *TCN's AUC per level*.

Por outro lado, analisou-se também que a medida que o número de *levels* (número de camadas) numa *TCN* aumenta, o seu desempenho melhora. Outro ponto que é importante realçar é que *AUC* de uma *CNN* + *Residual block* (Figura 5.8b) é bastante melhor do que a da *TCN* com *level* = *1* (Figura 5.9) apesar de as arquiteturas neste cenário serem bastante semelhantes. Esta diferença deve-se ao facto de que a *TCN* com apenas 1 nível não consegue ter em conta todo o sinal de entrada. Logo, esta não está a ser aplicada de forma adequada neste problema, pois a quantidade de níveis não é suficiente para tirar o proveito da sua principal vantagem que é a capacidade de ter em conta um histórico muito grande.

5.4.2 O impacto dos sinais artificiais (*DCGAN*) no treino das *CNNs*

Para além de treinar o modelo apenas com os dados reais e falsos, decidiu-se utilizar a abordagem da *DCGAN* para produzir dados novos (gerados) e verificar se é possível melhorar ainda mais a robustez de um classificador, tal como é descrito em diversos artigos que foram mencionados na Secção 2.5 do Trabalho Relacionado.

Deste modo, em primeiro lugar, esta subsecção surge com o objetivo de apresentar os resultados obtidos ao gerar sinais de pulsação falsos recorrendo a vários modelos de *DCGANs* com diferentes combinações de *hyperparameters*. Em segundo lugar, analisaram-se os resultados de diversos detetores de ataques após o treino com três tipos de dados diferentes: reais, falsos e gerados. Em que os dados Falsos e Gerados pertencem à mesma classe 1 (*Positive class*), uma vez que, ambos representam os sinais extraídos dos vídeos de ataque de apresentação. Por fim, retirou-se as devidas conclusões acerca do impacto da *DCGAN* no processo de aprendizagem dos algoritmos de deteção de vivacidade.

5.4.2.1 Convergência no treino da *DCGAN*

Antes de treinar os modelos de deteção de ataques de apresentação, houve a necessidade de treinar a *DCGAN* para que fosse possível obter novos sinais praticamente indistinguíveis dos reais. Para verificar se o modelo Gerador da *DCGAN* estava realmente a aprender a distribuição dos dados reais, decidiu-se criar a Figura 5.10 que representa o *loss error* durante o treino de ambos os modelos, do Discriminador e do Gerador.

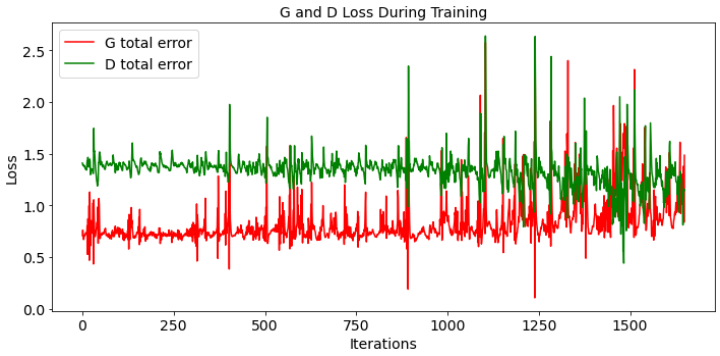


Figura 5.10: Convergência do *loss error* do Gerador e do Discriminador durante o treino da *DCGAN*.

Observando a figura conseguiu-se confirmar que ambos os modelos estão de facto a competir iterativamente ao longo do treino, acabando mesmo por convergirem. A produção desse efeito significa que o Gerador está progressivamente a evoluir e a conseguir gerar sinais de pulsação cada vez mais credíveis, sendo por isso, o *loss error* do Discriminador em alguns instantes aumenta e do Gerador diminui. Por outro lado, o Discriminador também está a melhorar o seu desempenho em classificar corretamente os sinais reais e os gerados, daí existirem os instantes em que o *loss error* do Discriminador diminui e do Gerador aumenta. Por causa dessas oscilações e melhorias constantes em ambos os modelos, conseguiu-se gerar sinais de pulsação falsos bastante realistas.

5.4.2.2 Geração de sinais de pulsação

Uma vez tendo as *DCGANs* treinadas, obteve-se um conjunto de dados novos falsos com uma distribuição bastante aproximada à dos dados reais. Com o intuito de demonstrar a comparação entre esses dois tipos de sinais, elaborou-se a Figura 5.11 que contém algumas partes da evolução da geração do sinal e inclusivé um sinal real que serve meramente para realizar a comparação.

Como é possível observar na figura, o último sinal gerado é extremamente parecido

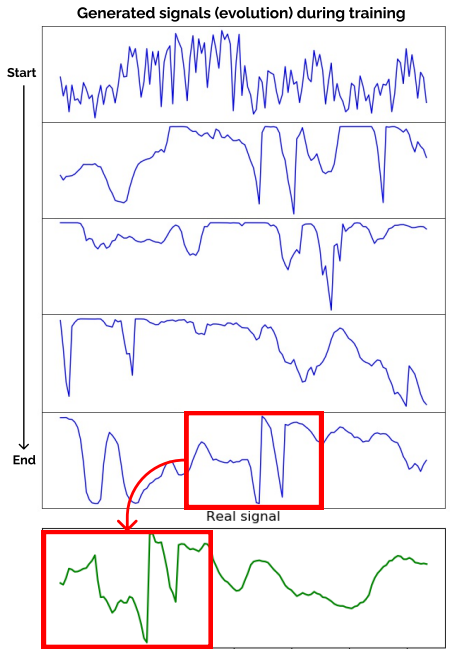


Figura 5.11: Evolução do sinal de pulsação gerado (linha azul) durante o treino da *DCGAN*. Em cima, o sinal inicial e em baixo, o sinal resultante.

com o sinal real que está representado a verde. Para além disso, analisou-se os sinais e assinalou-se a vermelho as partes do sinal real e do gerado que são bastante semelhantes. Desta forma, não se consegue ver a diferenciação a olho nu, mas o detector de ataques receberá essa informação. Portanto, fazendo o uso desses sinais gerados durante o treino dos detetores de ataques, espera-se melhorar a generalização e robustez e caso no futuro surjam ataques de apresentação cujo os sinais tenham características semelhantes, estes sejam capazes de os classificar como tal.

5.4.2.3 Robustez aos ataques de apresentação

Uma vez que se concluiu a recolha dos dados gerados, prosseguiu-se aos respetivos treinos dos detetores de ataques de apresentação. Treinos estes que consistem em três situações distintas: (1) treinar com os dados reais e os falsos (fotografias filmadas) (R *vs* F); (2) treinar com os dados reais e os gerados (R *vs* G); (3) treinar com os dados reais, falsos e os gerados (R *vs* $F + G$). Por outro lado, para poder comparar e analisar o comportamento dos detetores, decidiu-se ir aumentando progressivamente a arquitetura dos mesmos, começando com um classificador simples *Linear* e terminando com a *TCN*. Dito isto, elaborou-se a Tabela 5.7 que contém os resultados da execução do melhor modelo

de cada um dos detetores de ataques para cada tipo de conjunto de dados utilizados no processo de treino, validação e teste. As métricas de avaliação utilizadas são as mesmas que foram apresentadas na subsecção anterior.

Tabela 5.7: Resultados (em percentagem %) de deteção de ataques de apresentação em diferentes cenários. R - Real Live; F - Fake Live (fotografias filmadas); G - Generated Live (obtidos através da *DCGAN*). Melhoria aproximada representa o aumento no desempenho do modelo do cenário *R vs F* para *R vs F + G*.

Detetores de ataque	Cenário	Recall-1	F1-score-1	ROC AUC	Melhoria aprox.
Linear	R vs F	90,82	79,69	67,73	1,2%
	R vs G	75,68	80,99	89,07	
	R vs F + G	82,32	73,98	**68,55**	
DCGAN - 2 CNN layers	R vs F	63,02	73,37	81,13	7,9%
	R vs G	93,44	93,06	97,73	
	R vs F + G	86,00	84,11	**88,08**	
DCGAN(Original Attack Architecture)	R vs F	82,32	83,39	83,82	6,8%
	R vs G	92,90	94,84	98,06	
	R vs F + G	80,97	85,13	**89,97**	
DCGAN + 2 CNN layers	R vs F	83,67	84,99	87,37	5,1%
	R vs G	89,89	93,33	97,80	
	R vs F + G	84,29	87,39	**92,02**	
CNN+Res	R vs F	90,96	87,59	90,21	0,6%
	R vs G	94,54	95,71	98,23	
	R vs F + G	91,42	89,80	**90,73**	
TCN	R vs F	89,47	87,70	90,17	3,6%
	R vs G	95,08	95,47	98,49	
	R vs F + G	84,92	88,70	**93,55**	

Observando os resultados da Tabela 5.7 e Figura 5.12, conseguiu-se concluir que, **independentemente do detetor do ataque, o desempenho do mesmo melhora significativamente ao introduzir os sinais de pulsação gerados no conjunto de dados (Tabela 5.7 coluna Melhoria aprox.)**. Realizando a comparação mais específica entre os casos em que os dados utilizados foram *R vs F* e o *R vs F + G*, consegue-se perfeitamente notar o grande impacto na robustez dos modelos ao incluir os dados produzidos pelas *DCGANs*. Desta forma, entende-se que o algoritmo ao utilizar o conjunto de dados *R vs F + G*, torna-se mais robusto a distinguir as duas classes (maior *ROC AUC*) apesar do *Recall-1* piorar ligeiramente em comparação ao cenário *R vs F*. No entanto, tal como referido anteriormente, é necessário fazer um balanço entre essas duas métricas, pois o que se pretende é ter um modelo com boa capacidade de distinção entre ambas as classes, mas também, bastante resiliente a ataques de apresentação.

R vs F

Linear	Recall	F1-Score	Support
0	23,18	32,61	358
1	90,82	79,69	741

DCGAN-2	Recall	F1-Score	Support
0	81,84	63,35	358
1	63,02	73,37	741

DCGAN	Recall	F1-Score	Support
0	68,72	66,94	358
1	82,32	83,39	741

DCGAN+2	Recall	F1-Score	Support
0	72,63	70,37	358
1	83,67	84,99	741

CNN+Res	Recall	F1-Score	Support
Real	65,36	71,02	358
Fake	90,96	87,59	741

TCN	Recall	F1-Score	Support
0	69,83	72,89	358
1	89,47	87,70	741

R vs F + G

Linear	Recall	F1-Score	Support
0	35,82	43,67	698
1	82,32	73,98	1114

DCGAN-2	Recall	F1-Score	Support
0	70,49	73,11	698
1	86,00	84,11	1114

DCGAN	Recall	F1-Score	Support
0	85,24	79,07	698
1	80,97	85,13	1114

DCGAN+2	Recall	F1-Score	Support
0	86,25	81,63	698
1	84,29	87,39	1114

CNN+Res	Recall	F1-Score	Support
Real	60,29	64,06	340
Fake	91,42	89,80	1107

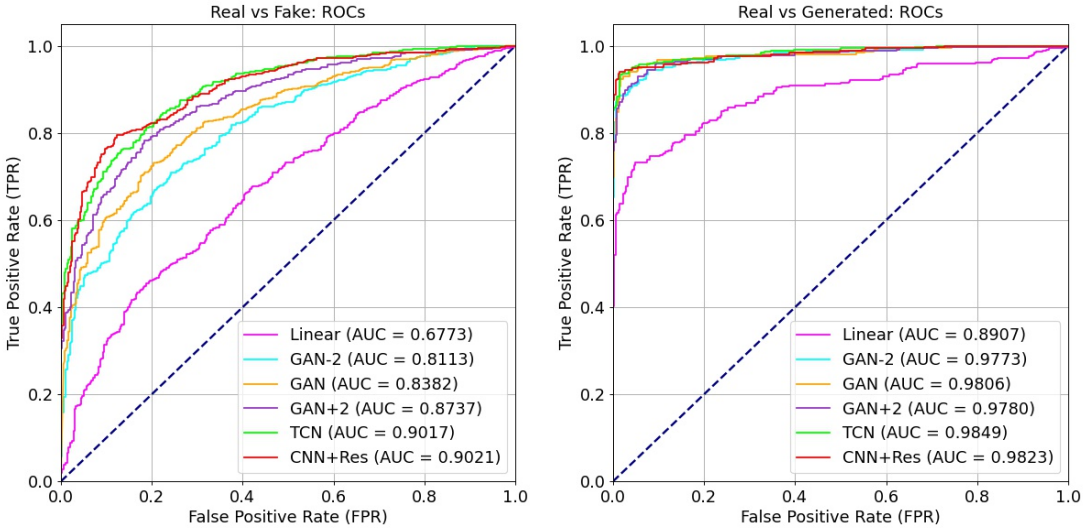
TCN	Recall	F1-Score	Support
0	89,54	83,84	698
1	84,92	88,70	1114

Figura 5.12: Comparação dos *classification reports* dos modelos apresentados na Tabela 5.7 cujo o conjunto de dados é *R vs F* e *R vs F + G*.

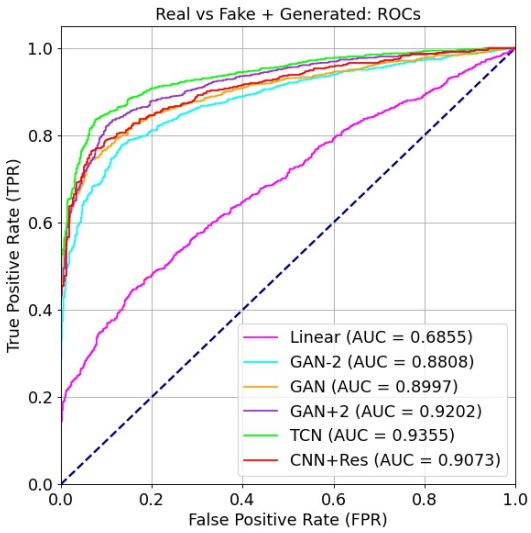
Para efetuar a comparação dos detetores de ataque em termos da *AUC*, concebeu-se a Figura 5.13, que representa as *ROC curves* de cada um deles para cada um dos conjuntos de dados utilizados.

Ao analisar a figura, vê-se que existe uma relação direta entre a complexidade do modelo e a *AUC*, uma vez que, a medida que à complexidade dos modelos aumenta, o valor da *AUC* aumenta de igual forma. Desta forma, sabendo que num cenário de vida real a arquitetura da *DCGAN* do atacante a partida não será conhecida, o modelo *TCN* conseguirá com bastante confiança (84,92) detetar os novos ataques e realizar a distinção entre as classes com uma certeza de (93,55).

Assim sendo, verificou-se que em ambas as experiências realizadas, o modelo que demonstra ser o mais apropriado para realizar a tarefa de deteção de ataques de apresentação é o modelo *TCN*, pois apesar de no priméiro cenário *R vs F* ter ficado ligeiramente atrás do modelo *CNN + Residual block*, neste novo cenário *R vs F + G* viu-se que apresentou os melhores resultados não só a distinguir ambas as classes (*AUC*) como também a



(a) *Real vs Fake* (b) *Real vs Generated*



(c) *Real vs Fake + Generated*

Figura 5.13: Comparação de todos os detetores de ataques utilizando a *AUC* nos respetivos conjuntos de dados.

detetar os sinais de pulsação reais (*Recall Reais* Tabela 5.12). Desta forma, concluiu-se que o facto de a *TCN* ter apresentado resultados ligeiramente inferiores do que *CNN + Residual block* no primeiro cenário poderá ter sido devido a quantidade reduzida de dados existentes.

Observando os resultados da Tabela 5.12 e a Figura 5.13 conseguiu-se concluír que

o facto de introduzir dados novos gerados recorrendo a *DCGAN* permite com que os modelos consigam melhorar o seu desempenho tanto ao nível de deteção de ataques de apresentação (*Recall-1*) como também na distinção entre as duas classes (*Positive* e *Negative*) de sinais de pulsação (*AUC*).

Como resultado desta avaliação dos modelos, concluiu-se que o melhor modelo é designado de *TCN* que contém uma característica bastante diferenciadora em relação aos outros modelos apresentados no Capítulo 3, uma vez que, permite que a classificação do sinal de pulsação tenha em conta todos os elementos do sinal (Figura A.1). Sendo por isso, a arquitetura da *TCN* é mais adequada e mais capaz para modelar os vários padrões necessários para a tarefa em questão.

Capítulo 6

Conclusões

Este trabalho apresenta uma abordagem inovadora que permite garantir uma maior segurança quando se trata de uma autenticação/registo a distância, ou seja, num ambiente não controlado como é o caso de efetuar o registo no sistema *SmartyFlow* que está integrado na plataforma da *Vision-Box*. Nesse sentido, recorrendo a deteção de vivacidade através dos sinais de pulsação estimados pelo *EVM*, foram propostos vários modelos *CNN* dos quais *TCN* se salientou em cumprir melhor os requisitos de detetar os ataques de apresentação.

No entanto, apesar de a *TCN* apresentar bons resultados, procurou-se que esta fosse ainda mais robusta a ataques de apresentação. Tendo esse objetivo em mente, desenvolveu-se com base em Treino Adversarial uma *DCGAN* que está descrita no Capítulo 4. Este método permitiu com que inúmeros sinais de pulsação falsificados fossem gerados e consequentemente utilizados no processo de aprendizagem do modelo. Tal como era previsto, esta técnica permitiu uma melhoria significativa no desempenho da *TCN*, conseguindo desta forma alcançar com sucesso o objetivo estabelecido.

Para que este estudo sobre as diferentes variações de *CNNs* e *DCGAN* aplicados aos dados unidimensionais fosse partilhado com a comunidade de investigadores, foram submetidos os seguintes 3 artigos:

1. Ruslan Padnevych, David Semedo, David Carmo, João Magalhães. *"1-D Convolutional Neural Networks for Robust On the Fly Face Liveness Detection"* (Capítulo 3)

2. Ruslan Padnevych, David Semedo, David Carmo, João Magalhães. *"Robust Face Liveness Detection with Deep Convolutional Generative Adversarial Networks"* (Capítulo 4)

3. David Semedo, David Carmo, Ruslan Padnevych, João Magalhães. *"Contact-free Airport Borders with Biometrics-on-the-Move"*

6.1 Impacto

Este trabalho contribui com dois modelos em que ambos contribuem para o objetivo final que é deteção de ataques de apresentação na fase de registo no sistema *SmartyFlow* da entidade *Vision-Box* (Figura 6.1).

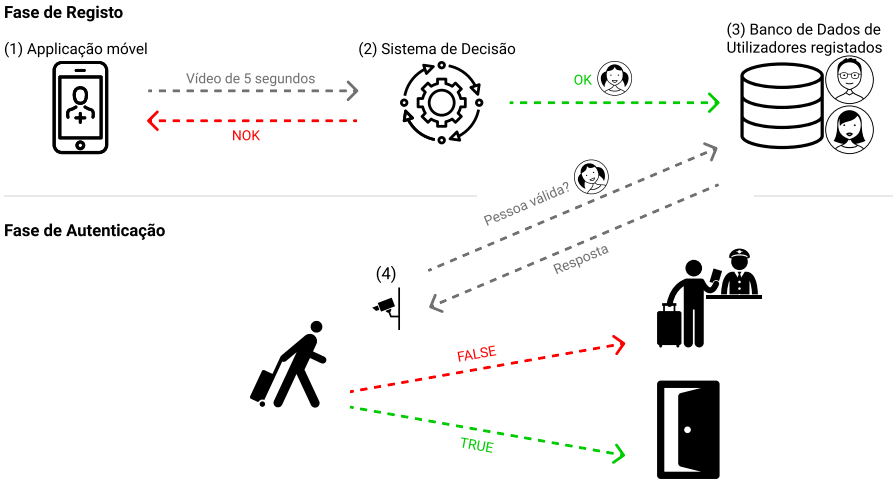


Figura 6.1: Diagrama de caso de uso – processo de registo e autenticação ao sistema de controlo de fronteiras nos aeroportos.

Ambos contribuem para que o **Sistema de Decisão** seja robusto ao ponto de garantir a segurança dos utilizadores. Um deles é o classificador (*TCN*) que tem como função distinguir os sinais de pulsação reais dos falsificados que serão extraídos no momento em que o utilizador envia um vídeo de 5s da sua face para proseguir com o registo. Os resultados do desempenho deste modelo que validam a sua preferência estão apresentados no Capítulo 5, exibindo uma capacidade em distinguir os sinais com uma *AUC* de 90,17 e uma robustez em detetar os ataques de apresentação de 89,47.

Por outro lado, uma vez que o número de exemplos de ataque são escassos, o desenvolvimento do segundo modelo (*DCGAN*) veio mesmo a propósito de resolver este

problema, permitindo gerar um número indefinido de sinais falsos, para que estes possam ser utilizados no treino do classificador *TCN* antes de este ser aplicado num cenário de vida real. Esta abordagem que é bastante rentável permitiu melhorar ainda mais a robustez do classificador destacado anteriormente. Como prova disso, também no Capítulo 5 estão os resultados da inclusão dessa técnica, uma vez que, o classificador *TCN* aumentou significativamente a sua capacidade de distinção dos sinais para 93,55. Além disso, melhorou também o desempenho na classificação dos sinais de pulsação genuínos (Reais).

Desta forma, tendo estes dois modelos desenvolvidos, foi possível criar um classificador bastante rápido a decidir e resiliente a ataques de apresentação. Tendo este valores de *Recall-1* e *AUC* suficientemente altos, concluiu-se que o algoritmo é capaz de impedir o impostor de tomar posse da identidade de um utilizador garantindo a sua segurança.

6.2 Trabalho Futuro

Uma vez que foi utilizada uma quantidade reduzida de dados durante o processo de treino, validação e teste dos modelos desenvolvidos, sugeria-se que, em primeiro lugar, as direções de trabalho futuro apontassem para a necessidade de criar um conjunto de dados muito mais extenso e diversificado, para que o modelo possa ser mais abrangente. Esta insuficiência deve-se, principalmente, às restrições impostas pelo contexto da pandemia (COVID-19) que estamos a viver neste momento. Sendo por isso, não foi possível reunir pessoas voluntárias dispostas a contribuir para este projeto.

Por outro lado, para além das combinações dos *hyperparameters* utilizados para realizar o afinamento dos modelos, outros valores podem ser explorados com finalidade de encontrar um modelo ainda mais robusto.

Fora as melhorias mencionadas, seria interessante estudar o porquê de os resultados de *AUC*, presentes no Capítulo 3 na Subsecção 5.4.1, entre a *CNN + Residual block* e a *TCN* com $level = 1$, serem bastante diferentes apesar de as arquiteturas serem muito semelhantes.

Por fim, visto que se utilizou um conjunto de abordagens distintas, como por exemplo, *EVM*, *TCN* e *DCGAN* de forma independente, apesar de, todas elas estarem relacionadas, faria todo o sentido a implementação de um sistema *end-to-end* ($Vídeo \rightarrow EVM \rightarrow (TCN + DCGAN) \rightarrow Result$), tal como está representado na Figura 6.2.

Desta forma, tendo apenas um só sistema de controlo, em que todos os algoritmos estariam interligados, os seus manuseamentos seriam muito mais práticos.

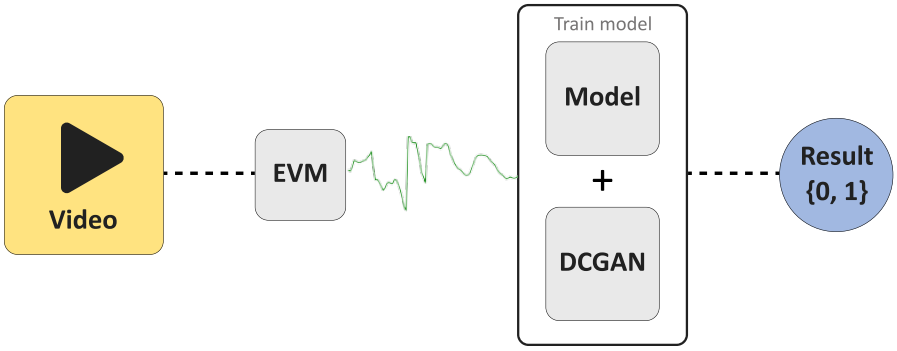


Figura 6.2: Representação do possível sistema *end-to-end*.

Bibliografia

[1] G. Abbas, M. Khan, R. Qureshi e K. Khurshid. "Scope of Video Magnification in Human Pulse Rate Estimation". Em: jan. de 2019, pp. 2–3.

[2] H. Abdenour, E. Nicholas, M. Sebastien e F. Julian. "Biometrics systems under spoofing attack: an evaluation methodology and lessons learned". Em: *IEEE Signal Processing Magazine* 32 (2015), p. 18.

[3] S. Bai, J. Z. Kolter e V. Koltun. "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling". Em: *CoRR* abs/1803.01271 (2018). URL: http://arxiv.org/abs/1803.01271.

[4] F. Bousefsaf, A. Pruski e C. Maaoui. "3D Convolutional Neural Networks for Remote Pulse Rate Measurement and Mapping from Facial Video". Em: *Applied Sciences* 9 (out. de 2019), p. 4364. DOI: 10.3390/app9204364.

[5] D. Canedo e A. Neves. "Facial Expression Recognition Using Computer Vision: A Systematic Review". Em: *Applied Sciences* 9 (nov. de 2019), p. 2. DOI: 10.3390/app9214678.

[6] R. Chellappa, G. Aggarwal e S. K. Zhou. "Face Recognition, Video-Based". Em: *Encyclopedia of Biometrics*. Ed. por S. Z. Li e A. Jain. Boston, MA: Springer US, 2009, pp. 366–372. ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_96. URL: https://doi.org/10.1007/978-0-387-73003-5_96.

[7] A. M. Delaney, E. Brophy e T. E. Ward. *Synthesis of Realistic ECG using Generative Adversarial Networks*. 2019. arXiv: 1909.09150 [eess.SP].

[8] C. Donahue, J. J. McAuley e M. S. Puckette. "Synthesizing Audio with Generative Adversarial Networks". Em: *CoRR* abs/1802.04208 (2018). eprint: 1802.04208. URL: http://arxiv.org/abs/1802.04208.

[9] P. Ekman e W. Friesen. *Facial Action Coding System*. vol. 1. Consulting Psychologists Press, 1978.

[10] J. Fan, S. Upadhye e A. Worster. "Understanding receiver operating characteristic (ROC) curves". Em: *Canadian Journal of Emergency Medicine* 8.1 (2006), 19–20. DOI: 10.1017/S1481803500013336.

[11] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger e H. Greenspan. "GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification". Em: *Neurocomputing* 321 (mar. de 2018). DOI: 10.1016/j.neucom.2018.09.013.

[12] J. Galbally, S. Marcel e J. Fierrez. "Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition". Em: *IEEE Transactions on Image Processing* 23.2 (fev. de 2014), pp. 710–724. ISSN: 1941-0042. DOI: 10.1109/TIP.2013.2292332.

[13] T. Golany, G. Lavee, S. Tejman Yarden e K. Radinsky. "Improving ECG Classification Using Generative Adversarial Networks". Em: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.08 (2020). DOI: 10.1609/aaai.v34i08.7037.

[14] I. Goodfellow, Y. Bengio e A. Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville e Y. Bengio. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin e A. C. Courville. "Improved Training of Wasserstein GANs". Em: *CoRR* abs/1704.00028 (2017). eprint: 1704.00028. URL: http://arxiv.org/abs/1704.00028.

[17] K. G. Hartmann, R. T. Schirrmeister e T. Ball. *EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals*. 2018. arXiv: 1806.01875 [eess.SP].

[18] K. He, X. Zhang, S. Ren e J. Sun. "Deep Residual Learning for Image Recognition". Em: *CoRR* abs/1512.03385 (2015). URL: http://arxiv.org/abs/1512.03385.

[19] J. Hernandez-Ortega, J. Fierrez, A. Morales e J. Galbally. "Introduction to Face Presentation Attack Detection". Em: abr. de 2019, pp. 4–7, 9–12. DOI: 10.1007/978-3-319-92627-8_9.

[20] G. Heusch e S. Marcel. "Pulse-based Features for Face Presentation Attack Detection". Em: out. de 2018, pp. 1–8. DOI: 10.1109/BTAS.2018.8698579.

[21] X. Hou, L. Shen, K. Sun e G. Qiu. "Deep Feature Consistent Variational Autoencoder". Em: *CoRR* abs/1610.00291 (2016). eprint: 1610.00291. URL: http://arxiv.org/abs/1610.00291.

[22] S. Iftikhar, R. Younas, N. Nasir e K. Zafar. "Detection and Classification of Facial Expressions using Artificial Neural Network". Em: *International Journal of Information Technology and Electrical Engineering* 3 (mar. de 2014), pp. 18–22.

[23] S. Ioffe e C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. arXiv: 1502.03167.

[24] M. Krišto e M. Ivašić-Kos. "An overview of thermal face recognition methods". Em: mai. de 2018, pp. 1098–1103. DOI: 10.23919/MIPRO.2018.8400200.

[25] A. Lagorio, M. Tistarelli, M. Cadoni, C. Fookes e S. Sridharan. "Liveness detection based on 3D face shape analysis". Em: abr. de 2013, pp. 2–4. ISBN: 978-1-4673-4987-1. DOI: 10.1109/IWBF.2013.6547310.

[26] "Temporal Characterization of Faces". Em: *Encyclopedia of Biometrics*. Ed. por S. Z. Li e A. Jain. Boston, MA: Springer US, 2009, pp. 1327–1327. ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_883. URL: https://doi.org/10.1007/978-0-387-73003-5_883.

[27] B. Lin, X. Li, Z. Yu e G. Zhao. "Face Liveness Detection by rPPG Features and Contextual Patch-Based CNN". Em: mai. de 2019, pp. 61–68. ISBN: 978-1-4503-6305-1. DOI: 10.1145/3345336.3345345.

[28] Y. Liu, A. Jourabloo e X. Liu. "Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision". Em: jun. de 2018, pp. 389–398. DOI: 10.1109/CVPR.2018.00048.

[29] E. Luz, D. Menotti e W. Schwartz. "Análise do Uso do Sinal de ECG em Baixas Frequências como Biometria". Em: ago. de 2013, p. 2.

[30] V. Mistry, J. J. Engelsma e A. K. Jain. "Fingerprint Synthesis: Search with 100 Million Prints". Em: *CoRR* abs/1912.07195 (2019). eprint: 1912.07195. URL: http://arxiv.org/abs/1912.07195.

[31] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior e K. Kavukcuoglu. "WaveNet: A Generative Model for Raw Audio". Em: *CoRR* abs/1609.03499 (2016). eprint: 1609.03499. URL: http://arxiv.org/abs/1609.03499.

[32] A. Radford, L. Metz e S. Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: 1511.06434.

[33] R. Raghavendra e C. Busch. "Presentation Attack Detection Methods for Face Recognition Systems - A Comprehensive Survey". Em: *ACM Computing Surveys* 50 (jan. de 2017), pp. 6, 10–15. DOI: 10.1145/3038924.

[34] M. Roomi e D. Beham. "A Review Of Face Recognition Methods". Em: *International Journal of Pattern Recognition and Artificial Intelligence* 27 (abr. de 2013), pp. 2–3, 5, 13, 19. DOI: 10.1142/S0218001413560053.

[35] P. Sajida, M. S. A. Sharifah, H. A. Nidaa, A. W. A. Wan, H. Marsyita e N. Nadeem. "Face Liveness Detection Using Dynamic Local Ternary Pattern (DLTP)". Em: *Computers* (2016), p. 2.

[36] R. Sanchez-Reillo. "Hand Geometry". Em: *Encyclopedia of Biometrics*. Ed. por S. Z. Li e A. Jain. Boston, MA: Springer US, 2009, pp. 677–682. ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_251. URL: https://doi.org/10.1007/978-0-387-73003-5_251.

[37] J. Seo e I.-J. Chung. "Face Liveness Detection Using Thermal Face-CNN with External Knowledge (SCIE)". Em: *Symmetry* 11 (mar. de 2019), pp. 1–2. DOI: 10.3390/sym11030360.

[38] M. Serhii. "Anti-spoofing techniques in face recognition". Em: (ago. de 2019), p. 5.

[39] I. O. for Standardization. "Information technology — Biometric presentation attack detection — Part 1: Framework". Em: (jan. de 2016), pp. 5, 7–9, 11–13.

[40] I. O. for Standardization. "Information technology — Vocabulary — Part 37: Biometrics". Em: (fev. de 2017), pp. 7–8.

[41] M. Sund Levander, C. Forsberg e P. MLT. "Normal oral, rectal, tympanic and axillary body temperature in adult men and women: A systematic literature review". Em: *Scandinavian Journal of Caring Sciences* 16 (mai. de 2002), pp. 122 –128. DOI: 10.1046/j.1471-6712.2002.00069.x.

[42] S. Tirunagari, N. Poh, D. Windridge, A. Iorliam, N. Suki e A. Ho. "Detection of Face Spoofing Using Visual Dynamics". Em: *Information Forensics and Security, IEEE Transactions on* 10 (abr. de 2015), pp. 762–777. DOI: 10.1109/TIFS.2015.2406533.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser e I. Polosukhin. "Attention Is All You Need". Em: *CoRR* abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762.

[44] M. Wagner e G. Chetty. "Liveness Assurance in Face Authentication". Em: *Encyclopedia of Biometrics*. Ed. por S. Z. Li e A. Jain. Boston, MA: Springer US, 2009, pp. 908–916. ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_67. URL: https://doi.org/10.1007/978-0-387-73003-5_67.

[45] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand e W. T. Freeman. "Eulerian Video Magnification for Revealing Subtle Changes in the World". Em: *ACM Transactions on Graphics (Proc. SIGGRAPH 2012)* 31.4 (2012).

[46] C. Xie, J. Wang, Z. Zhang, Z. Ren e A. L. Yuille. "Mitigating adversarial effects through randomization". Em: *CoRR* abs/1711.01991 (2017). URL: http://arxiv.org/abs/1711.01991.

[47] A. Yousef, L. Yaojie, J. Amin e L. Xiaoming. "Face anti-spoofing using patch and depth-based CNNs". Em: *2017 IEEE International Joint Conference on Biometrics (IJCB)* (2017), pp. 319–328.

[48] F. Zhu, F. Ye, Y. Fu, Q. Liu e B. Shen. "Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network". Em: *Scientific Reports* 9.6734 (2019). DOI: 10.1038/s41598-019-42516-z.

Webgrafia

[49] *Digital Security Magazine*. Acedido em: 2020-02-01. URL: https://www.digitalsecuritymagazine.com/en/2013/05/08/la-videovigilancia-biometrica-toma-una-nueva-dimension-con-biosurveillance-next-de-herta-security/.

[50] FINDBIOMETRICS. *Standardized Testing for Biometrics: Cutting Through the Hype and Finding Integrity in Digital Identity*. Acedido em: 2020-01-23. Ago. de 2019. URL: https://findbiometrics.com/standardized-testing-for-biometrics-cutting-through-the-hype-and-finding-integrity-in-digital-identity-facetec-white-paper/3/.

[51] FRONTEX. *Frontex testing the future of border checks at Lisbon airport*. Acedido em: 2020-01-21. Out. de 2019. URL: https://frontex.europa.eu/media-centre/news-release/frontex-testing-the-future-of-border-checks-at-lisbon-airport-DI84r4.

[52] A. Hauck. *Different Types of Biometrics*. Acedido em: 2020-01-21. Set. de 2019. URL: https://www.ibeta.com/different-types-of-biometrics/.

[53] *IndiaMART*. Acedido em: 2020-02-01. URL: https://www.indiamart.com/proddetail/face-recognition-based-attendance-system-17531072573.html.

[54] N. Inkawhich. *DCGAN Tutorial*. Acedido em: 2020-10-02. Ago. de 2020. URL: https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html.

[55] B. Institute. *Types of Biometrics*. Acedido em: 2020-01-21. 2020. URL: https://www.biometricsinstitute.org/what-is-biometrics/types-of-biometrics/.

[56] S. Mayhew. *Explainer: Verification vs. Identification Systems*. Acedido em: 2020-01-22. URL: https://www.biometricupdate.com/201206/explainer-verification-vs-identification-systems.

[57] M. Rouse. *Adversarial Machine Learning*. Acedido em: 2020-10-20. Jul. de 2019. URL: https://searchenterpriseai.techtarget.com/definition/adversarial-machine-learning.

Apêndice A

TCN: Previsão do último elemento

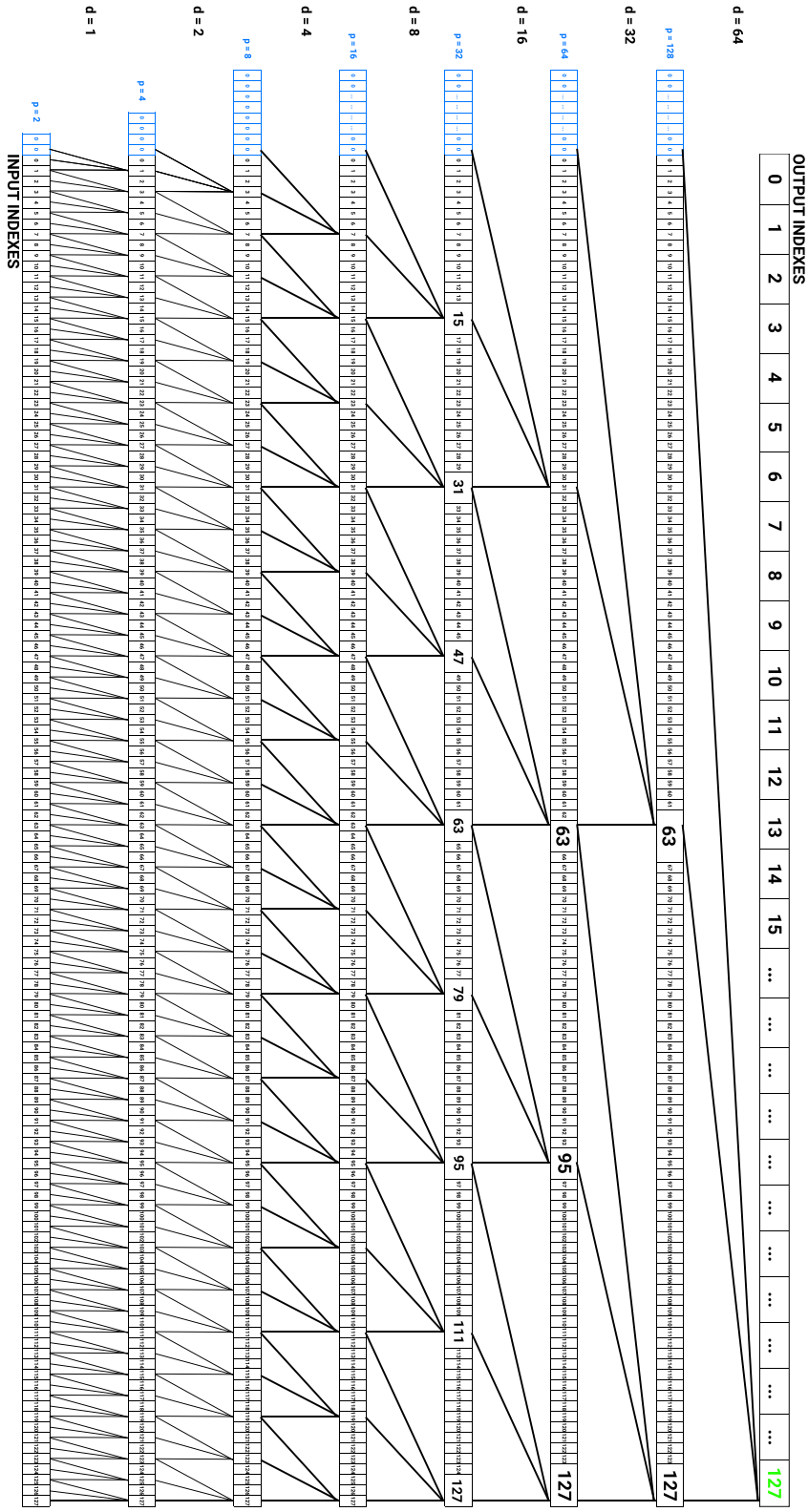


Figura A.1: Previsão do último elemento da sequência ($\hat{y}_{127}$) tendo em conta todos os elementos do sinal de pulsação.

ANEXO I

TCN como Discriminador da *DCGAN*

Para além da abordagem em treinar a *DCGAN* com o Discriminador original, realizou-se também a experiência em que no lugar desse Discriminador se colocou a *TCN*. O objetivo desta substituição era verificar se ao colocar um Discriminador mais complexo era possível ou não produzir sinais gerados muito mais plausíveis. Como resultado desta experiência obteve-se a Figura I.1 que representa o progresso do *loss error* ao longo do treino da *DCGAN*.

Figura I.1: A evolução do *loss error* durante o treino da *DCGAN*.

Ao analisar o gráfico, concluiu-se que quando o Discriminador tem uma arquitetura bastante mais complexa do que a do Gerador, é de esperar que, a partir de um certo ponto no treino, o Gerador fica em crescente desvantagem, pois o Discriminador consegue facilmente distinguir ambas as classes.